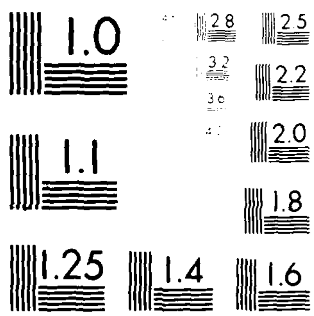
AD-A110 334   GENERAL MOTORS CORP INDIANAPOLIS IN DETROIT DIESEL A--ETC F/6 21/5
EXPERIMENTAL INVESTIGATION OF TURBINE ENDWALL HEAT TRANSFER. VO--ETC(U)
SEP 81 L D HYLTON, M S MIHELC, E R TURNER   F33615-77-C-2030
UNCLASSIFIED DDA-EDR-10363-VOL-3   AFWAL-TR-81-2077-VOL-3   NL

1 OF 1
AD A
110 334

END
DATE
FILMED
02-82
DTIC



MICROCOPY RESOLUTION TEST CHART

AFWAL-TR-81-2077
Volume III

EXPERIMENTAL INVESTIGATION OF TURBINE ENDWALL HEAT TRANSFER

Volume III. Data Base System

L. D. Hylton, M. S. Mihelc, E. R. Turner, and R. E. York

Detroit Diesel Allison
Division of General Motors Corporation
P. O. Box 894
Indianapolis, Indiana 46206

September 1981

FINAL REPORT for period 1 August 1977 - 31 August 1981

Approved for public release; distribution unlimited

AERO PROPULSION LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Kervyn D Mach
KERVYN D. MACH
Project Engineer

Charles E. Bentz
CHARLES E. BENTZ, Tech Area Manager
Hot Section Technology Group

FOR THE COMMANDER

David H Quick
DAVID H. QUICK, Lt Col, USAF
Deputy Director, Turbine Engine Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/POTC, *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWAL-TR-81-2077, Vol III	2. GOVT ACCESSION NO.: AD-A110 331
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): EXPERIMENTAL INVESTIGATION OF TURBINE ENDWALL HEAT TRANSFER (VOL. III, DATA BASE SYSTEM)	5. TYPE OF REPORT & PERIOD COVERED: Final 1 Aug 77 - 31 Aug 81
	6. PERFORMING ORG. REPORT NUMBER: EDR 10363, Vol. III
7. AUTHOR(s): L. D. Hylton, M. S. Mihelc, E. R. Turner, R. E. York	8. CONTRACT OR GRANT NUMBER(s): F33615-77-C-2030
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Detroit Diesel Allison, Division of General Motors Corporation, Indianapolis, Indiana 46206	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62203F, 3066 06 30
11. CONTROLLING OFFICE NAME AND ADDRESS: Aero Propulsion Laboratory (AFWAL/POTC), Wright-Patterson AFB, Ohio 45433	12. REPORT DATE: September 1981
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Turbine Heat Transfer, Turbine Secondary Flow, Turbine Endwall Heat Transfer, Turbine Cascades, Heat Transfer Data Base

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two turbine cascades were tested at simulated engine conditions to provide a data base of endwall heat transfer data. This data base is intended to be sufficiently complete to provide verification data for refined computational models developed to predict first-stage stator endwall heat transfer in advanced turbine engines.

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 049200

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Cont)

A linear, two-dimensional cascade provided the bulk of the data. This cascade provided data to separate the effects of exit Mach number, exit Reynolds number, inlet boundary layer thickness, gas-to-wall temperature ratio, inlet pressure gradients, and inlet temperature gradients. In addition, adiabatic wall temperature and inlet turbulence intensity data are available for the linear cascade runs. A computerized data base was generated. This data base, with its associated software management system, provides the user with relatively easy access to the vast amount of data generated.

A full annular, three-dimensional cascade was used to acquire data for identifying the radial pressure gradient effects. Tests in the annular cascade were run over a wide range of exit Mach and Reynolds numbers and gas-to-wall temperature ratios, all at levels typical of advanced engines.

TABLE OF CONTENTS

VOLUME III

LIST OF ILLUSTRATIONS

VOLUME III

LIST OF TABLES

VOLUME III

1.0 INTRODUCTION

The purpose of the heat transfer data base system is to organize the massive data base generated under Contract F33615-77-C-2030 and to provide a convenient way of accessing the data through the computer. The data base system solves the problem of storing large volumes of printed data, since data are stored on magnetic tape. The data base is read from tape into a direct-access disk file that may be accessed in a number of ways using the software routines provided. The direct computerized access avoids the human errors associated with transferring large quantities of data from printed sheets to punched cards or other computer input formats.

Figure 1 is a schematic of the endwall heat transfer data base system at Detroit Diesel Allison. The figure highlights the parts of the data base delivered and described in this volume.

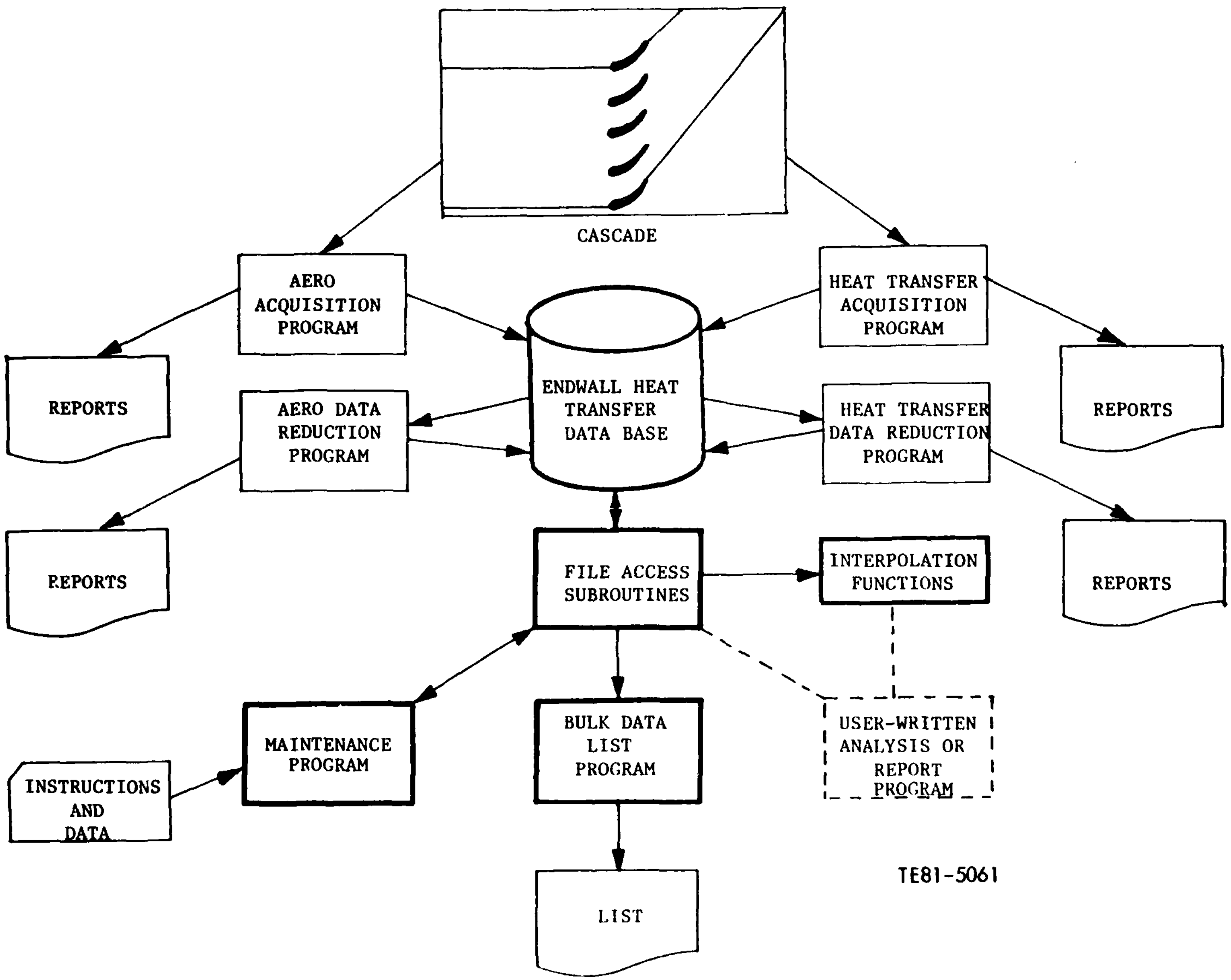


Figure 1. Endwall heat transfer data base system.

2.0 SYSTEM OVERVIEW

2.1 DATA CONTENT

The contents of the data base include all raw and reduced data and geometry information used to generate the results reported in Volume II of this report. All temperatures are given in °F, all pressures in psia, and all dimensions (including relative coordinate locations) in inches. Velocities are given in ft/sec, and Reynolds numbers are based on true chord. Any other unique dimensions are stored in the list program and may be obtained from an index listing.

The listing program specifies the basis of several derived quantities in terms of subscripted variable names. Table I shows the subscript convention and defines the subscripted variable names. Definitions for the derived loss parameters also are included in Table I.

TABLE I. SUBSCRIPTED VARIABLE DEFINITIONS

Subscript convention: 1--measured inlet
2--measured exit
X--ideal exit
3--uniform mixed-out exit

Variable	Definition
PT1	Measured inlet total pressure
PS1	Average of measured inlet sidewall static pressures
TT1	Measured inlet total temperature
PTX	Ideal exit total pressure (= PT1)
PSX	Average of measured exit sidewall static pressures
TTX	Ideal exit total temperature (= TT1)
PT2	Exit cone probe total pressure
PS2	Exit cone probe static pressure
TT2	Exit probe total temperature
OMEGA2	(PT1-PT2)/(PTX-PSX)
DSOCP2	$\mathrm{Ln}\ (TT2/TT1) - \left(\frac{\gamma-1}{\gamma}\right) \mathrm{Ln}\ (PT2/PT1)$
EBAR2	$\dfrac{(PTX/PT2)^{\frac{\gamma-1}{\gamma}} - 1.}{(PTX/PS2)^{\frac{\gamma-1}{\gamma}} - 1.}$
PT3	Calculated mixed-out total pressure
PS3	Calculated mixed-out static pressure
OMEGA3	(PTX--PT3)/(PTX--PS3)
DSCP3	$\mathrm{Ln}\ (TTX/TT1) - \left(\frac{\gamma-1}{\gamma}\right) \mathrm{Ln}\ (PT3/PT1)$
EBAR3	$\dfrac{(PTX/PT3)^{\frac{\gamma-1}{\gamma}} - 1.}{(PTX/PS3)^{\frac{\gamma-1}{\gamma}} - 1.}$

Most of the raw data items stored in the data base have been reviewed and edited to flag obvious bad readings due to instrumentation failures. The edited readings were flagged by replacing the bad data with the value -1. All thermocouple readings for the heat transfer endwall have been edited to include only those readings used as input to the finite element analysis program. Thermocouple data from the adiabatic endwall, the vane, and upstream total temperature rakes were also edited. Upstream total and static pressures, endwall static pressures, and exit static pressures were edited as well.

Data associated with the boundary layer pressure and temperature rakes were not edited. The data from these rakes were reliable for early runs, but indicated significant instrumentation failures as time went on. Since data from the traversing inlet probe have been supplied to provide boundary layer information, the boundary layer rake data were not edited. These data have been included in the data base, however, and should therefore be used with caution.

2.2 DATA ORGANIZATION

The data base consists of sets of data, and each set includes all data with a common identifier. The present data base consists of two types of data sets, but may be expanded as the user desires.

The first type of data set in the data base is the single data set GEOMETRY. This set includes relative coordinates for all endwall instrumentation and vane thermocouples. The data set also includes a vane profile description. Each item included in the set GEOMETRY has an assigned identification number that is used in accessing the data. Table II lists each GEOMETRY variable and its identification (I.D.) number.

TABLE II. DATA SET GEOMETRY VARIABLE I.D. MAP

ID NO.	DESCRIPTION	DIMENSIONS PRIMAR.	DIMENSIONS SECONDARY
1	H.T. ENDWALL HOT SIDE T/C COORDINATES	2	53
2	H.T. ENDWALL COLD SIDE T/C COORDINATES	2	31
3	FINITE ELEMENT MODEL NODE COORDINATES	2	338
4	ADIABATIC ENDWALL T/C COORDINATES	2	90
5	ENDWALL STATIC PRESSURE COORDINATES	2	42
6	VANE T/C COORDINATES	2	113
7	ENDWALL BOUNDARY POINTS	3	85
8	VANE PROFILE COORDINATES	2	60
9	VANE TRUE CHORD	1	1
10	VANE SPACING	1	1
11	VANE LEADING EDGE RADIUS	1	1
12	VANE TRAILING EDGE RADIUS	1	1
13	VANE SETTING ANGLE	1	1
14	CASCADE THROAT HEIGHT	1	1

The second type of data set currently stored in the data base is made up of all data associated with a particular experimental run number. Forty-four such data sets, RUNXXXXX, are stored in the data base. Each variable stored for a RUNXXXXX set has an associated identification number. The general map of variables and associated identification numbers used for all RUNXXXXX data sets is shown in Table III. See Section 3 of Volume II of this report to determine the run conditions and data available for a given run number. A detailed index of the variables stored for each run number may also be generated using the program BLIST.

2.3 MANAGEMENT SOFTWARE

The management software provided with the data base includes two stand-alone programs for editing and listing data, two interpolation function subroutines, and 18 general access subroutines for data item retrieval and storage.

The stand-alone programs are MAINT and BLIST. These programs use the access subroutines to perform routine tasks. MAINT facilitates the interrogation of the data base with the option of editing, creating, or deleting data sets. BLIST provides for listing all or any subset of the data base.

The interpolation functions have been provided for the analyst to facilitate use of the data base. ANTRP provides a single value for any variable associated with the endwall region for a given (x,y). BNTRP returns a single value of any variable associated with the traversing exit probe for a given (z,y) point in the defined exit plane. The interpolation functions may be incorporated in any user-written program accessing the data base.

The file access subroutines are the building blocks for programs interacting with the data base. They are generalized subroutines for item storage and retrieval. The access subroutines are written in standard FORTRAN, and may be incorporated in any program requiring data base interaction.

TABLE III. DATA SET RUNXXXXX VARIABLE I.D. MAP

ID NO.	DESCRIPTION	DIMENSIONS PRIMARY	DIMENSIONS SECONDARY
1	DATE: MO, DAY, YEAR	3	1
2	INLET TOTAL TEMPERATURE	1	1
3	IDEAL EXIT MACH NUMBER	1	1
4	INLET TOTAL PRESSURE	1	1
5	AVERAGE INLET STATIC PRESSURE	1	1
6	AVERAGE EXIT STATIC PRESSURE	1	1
7	INLET REYNOLDS NUMBER/ 10E6	1	1
8	IDEAL EXIT REYNOLDS NUMBER/ 10E6	1	1
9	AVERAGE T_{WALL}/T_{GAS}	1	1
10	T14 TRANSITION SECTION WALL TEMP.	1	1
11	CASCADE MASS FLOW RATE	1	1
12	INLET MACH NUMBER	1	1
13	EXPANSION RATIO	1	1
14	PRESSURE RATIO	1	1
15	INLET V/VCRITICAL	1	1
16	EXIT V/VCRITICAL	1	1
17			
18			
19			
20			
21			
22			
23			
24			
25	CORE RAKE PRESSURES	5	2
26	CORE RAKE TEMPERATURES	5	2
27	SIDEWALL STATIC PRESSURES - INLET	9	1
28	SIDEWALL STATIC PRESSURES - EXIT ROW1	9	1
29	SIDEWALL STATIC PRESSURES - EXIT ROW2	13	1
30	BOUNDARY LAYER RAKE PRESSURES	16	2
31	BOUNDARY LAYER RAKE TEMPERATURES	16	1
32	ENDWALL STATIC PRESSURES	42	1
33	VANE STATIC PRESSURES -PRESSURE SIDE	36	1
34	VANE STATIC PRESSURES -SUCTION SIDE	30	1
35			
36			
37	INLET TRAVERSE PROBE LENGTH	1	1
38	INLET TRAVERSE PROBE WIDTH	1	1
39	INLET TRAVERSE EXTERIOR ANGLE	1	1

ID NO.	DESCRIPTION	DIMENSIONS PRIMARY	DIMENSIONS SECONDARY
40	INLET TRAVERSE DISTANCE FROM WALL	36	1
41	INLET TRAVERSE CORRECTED PRESSURE	36	1
42	INLET TRAVERSE TEMPERATURE	36	1
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60	EXIT PROBE SPANWISE POSITION	9	1
61	EXIT PROBE VERTICAL POSITION	25	9
62	EXIT PROBE REF. PRESSURE	25	9
63	EXIT PROBE TOTAL PRESSURE	25	9
64	EXIT PROBE STATIC PRESSURE 1	25	9
65	EXIT PROBE STATIC PRESSURE 2	25	9
66	EXIT PROBE STATIC PRESSURE 3	25	9
67	EXIT PROBE STATIC PRESSURE 4	25	9
68	EXIT PROBE TEMPERATURE	25	9
69	EXIT PROBE ANGLE	1	1
70			
71			
72			
73	GAS TEMPERATURE DURING ENDWALL READING	1	1
74	H.T. ENDWALL HOT SIDE TEMPERATURES	53	1
75	H.T. ENDWALL COLD SIDE TEMPERATURES	31	1
76	GAS TEMPERATURES DURING VANE T/C READING	1	1
77	VANE TEMPERATURES	113	1
78	ADIABATIC ENDWALL TEMPERATURE	90	1

TABLE III. (CONT)

ID NO.	DESCRIPTION	DIMENSIONS PRIMARY	DIMENSIONS SECONDARY
79	INSULATION TEMPERATURE 1	1	1
80	INSULATION TEMPERATURE 2	1	1
81	GAGE BLOCK TEMPERATURE 1	1	1
82	GAGE BLOCK TEMPERATURE 2	1	1
83			
84			
85	INLET TRAVERSE MACH NUMBER	36	1
86	INLET TRAVERSE VELOCITY	36	1
87			
88			
89			
90	EXIT TOTAL PRESSURE	25	9
91	EXIT STATIC PRESSURE	25	9
92	EXIT MACH NUMBER	25	9
93	EXIT YAW ANGLE	25	9
94	EXIT PITCH ANGLE	25	9
95	EXIT ZP (PASSAGE LOCATION)	25	9
96	EXIT V/V*	25	9
97	EXIT V/V* BASED ON PTX, PSPROBE	25	9
98	EXIT BETA2	25	9
99	EXIT OMEGA	25	9
100	EXIT EBAR	25	9
101	EXIT DS/CP	25	9
102	EXIT MASS FLUX	25	9
103	EXIT VELOCITY	25	9
104	EXIT AXIAL VELOCITY	25	9
105	EXIT TANGENTIAL VELOCITY	25	9
106	EXIT SPANWISE VELOCITY	25	9
107	EXIT TOTAL TEMPERATURE AT CONE PROBE	25	9
108			
109	EXIT AVERAGED MASS FLOW	9	1
110	EXIT AVERAGED TOTAL PRESSURE	9	1
111	EXIT AVERAGED STATIC PRESSURE	9	1
112	EXIT AVERAGED MACH NUMBER	9	1
113	EXIT AVERAGED YAW ANGLE	9	1
114	EXIT AVERAGED PITCH ANGLE	9	1
115	EXIT AVERAGED PT/PS	9	1
116	EXIT AVERAGED V/VCRITICAL	9	1
117	EXIT AVERAGED V/V* BASED ON PTX,PSPROBE	9	1

ID NO.	DESCRIPTION	DIMENSIONS PRIMARY	DIMENSIONS SECONDARY
118	EXIT AVERAGED BETA2	9	1
119	EXIT AVERAGED (PTX-PT)	9	1
120	EXIT AVERAGED (PTX-PT)/PTX	9	1
121	EXIT AVERAGED OMEGA	9	1
122	EXIT AVERAGED EBAR	9	1
123	EXIT AVERAGED DS/CP	9	1
124	EXIT AVERAGED PS2/PS1	9	1
125	EXIT AVERAGED V2	9	1
126	EXIT AVERAGED V2 AXIAL	9	1
127	EXIT AVERAGED V2 TANGENTIAL	9	1
128	EXIT AVERAGED V2 SPANWISE	9	1
129	EXIT AVERAGED T2 AT CONE PROBE	9	1
130	EXIT MIXED OUT PT3	9	1
131	EXIT MIXED OUT PS3	9	1
132	EXIT MIXED OUT XPAN RATIO	9	1
133	EXIT MIXED OT PS2/PS1	9	1
134	EXIT MIXED OUT MACH NUMBER	9	1
135	EXIT MIXED OUT BETA3	9	1
136	EXIT MIXED OUT OMEGA3	9	1
137	EXIT MIXED OUT EBAR3	9	1
138	EXIT MIXED OUT DS/CP	9	1
139	EXIT MIXED OUT V/V*3	9	1
140	EXIT MIXED OUT V/V* BASED ON PTX, PS3	9	1
141	EXIT MIXED OUT V3	9	1
142	EXIT MIXED OUT V3 AXIAL	9	1
143	EXIT MIXED OUT V3 TANGENTIAL	9	1
145	EXIT AVERAGE MASS FLUX	1	1
146	EXIT AVERAGE PT3	1	1
147	EXIT AVERAGE TT3	1	1
148	EXIT AVERAGE MACH NUMBER	1	1
149	EXIT AVERAGE BETA3	1	1
150	EXIT AVERAGE OMEGA3	1	1
151	EXIT AVERAGE EBAR3	1	1
152			
153			
154			
155			
156			
157	H.T. ENDWALL NODAL TEMPERATURES	338	1
158	H.T. ENDWALL NODAL HEAT FLUX	338	1
159	H.T. ENDWALL NODAL STANTON NUMBER	338	1

3.0 USER'S MANUAL

3.1 SOFTWARE DESCRIPTIONS/USAGE INSTRUCTIONS

3.1.1 Main Programs

The main programs supplied with the data base are a maintenance program and a bulk data listing program. Both are written to read the data base from magnetic tape and create a random-access data bank disk file. Both programs read input data from punched cards and output information to a line printer in 133-character records. The general input deck structure is outlined in Section 3.2. The programs interact with the random-access data bank, processing a series of commands. The maintenance program creates a new data base magnetic tape file.

Program 1--Maintenance (MAINT)

The purpose of the maintenance program, MAINT, is to provide a means of making minor changes in the data base. It is not intended to be used for entering large volumes of data. The program generates a printed record of changes made as an audit trail, listing old and new values. The program also permits inquiry about the value of any data item without updating it.

The maintenance program begins by reading a sequential tape file of the data base and creating a random-access data bank disk file. The program then communicates only with the random-access data bank, processing a series of input commands sequentially. When the command list is exhausted, the data bank disk file is closed and copied to a new sequential tape file, forming the new data base.

Input to MAINT consists of a series of commands using key words that may be arranged to suit the user's needs. The format of the commands is shown in Figure 2. Each command line is 30 characters long. Key words are underlined, and when shown as beginning in column one, they must appear there. When blanks are underlined, they are an integral part of the key word.

All numbers are read in free format. Decimal points and exponential notations may be used, as in the FORTRAN language. Repeated values may be coded as N * #.##, where N is the number of occurrences. All free-format numbers must be separated by blanks or commas. Data may be continued on subsequent lines, and may begin on the line following the identification specification. No number field may be broken by a continuation.

The commands shown in Figure 2 can be separated into four major types. The mode command specifies the function to be performed, the set command specifies the data set to be accessed, the I.D. commands specify the data manipulation within the data set, and the end command signals the end of the maintenance session.

```
1  INQUIRY

2  UPDATE
                        [CREATE]
3  SET NNNNNNNN         [DELETE]
                        [CREATE]
4  RUN ###              [DELETE]

5  ID ##   ##.##E±##

6  ID ## (##) ###.##

7  ID ## (*) ##.# ##  #.##  .## . . .

8  ID ###  (##,##)  ###

9  ID ##  (*,##)  # # # . . .

10 ID ##  (##,*)  ## ## ## . . .

11 ID ##  (*,*)  #.#,##,# . . .

12 ID ## ID ## ID ## (*) # # #  ## ## ##  . . .

13 ID ##  DELETE

14 ID ## ID ## ID ## . . .  DELETE

15 END
```

TE81-5062

Figure 2. Maintenance program commands.

The mode command specifies the function to be performed by subsequent commands. In the UPDATE mode, the data base is changed as dictated by following commands. The program begins in the UPDATE mode. The INQUIRY mode will return the data stored under the identification following.

The mode may be changed at any point in the run, and the program remains in that mode until another mode command is encountered. In the INQUIRY mode, the program ignores any data encountered.

The second major type of command is used to specify the data set under which the following identification numbers are to be found or inserted. The commands are numbered 3 and 4 in Figure 2. Data stored in the data base are divided into data sets. All data for a given experimental run number are stored under a data set called RUNXXXXX, where the Xs represent the run number. Instrumentation coordinates and other geometry information are stored in a set called GEOMETRY. The SET command is used to enter the eight-character set name directly. The RUN command synthesizes the eight-character set name for the free-format run number given, and then performs the same action as the SET command. One of the option key words may be included (beyond column 12 on the command line) with either of these commands. DELETE causes the entire set, including all of its data, to be deleted from the data base. CREATE must be present on the set command line (beyond column 12) if a new data set is to be initiated. The default action of the SET command is designed to use a set if it exists, and to print an error message if it does not, unless CREATE is specified.

The third major type of command, the I.D. command, is shown on lines 5 through 14 of Figure 2. The I.D. command is used to update or inquire about the value of an individual data item within a data set, depending on the most recent mode specification. The item specified may be a scalar, an entire vector, an element of a vector, an entire matrix, an element of a matrix, a row of a matrix, or a column of a matrix.

When the program is in the UPDATE mode, the list of new data is entered according to the identification number and subscript specification, as shown. Since the list may be continued on subsequent lines, the program will not consider the command to be complete until it sees the beginning of the next command. If the program is running in the INQUIRY mode, any data list erroneously supplied will be ignored.

Commands numbered 5 through 11 in Figure 1 illustrate the various data manipulations possible using MAINT. Command 5 shows how a scalar would be entered in the UPDATE mode. In the INQUIRY mode, no data should be present on the command line. Command 6 shows how to refer to an element of a vector, with the subscript placed in parentheses, as shown. This command is normally used for vectors that already exist in the data base. If a new vector is being created, however, it will be initialized with maximum dimensions equal to the subscript, and the last element will be assigned the data supplied.

Command 7 shows how to update or interrogate an entire vector. The parentheses and asterisk are not really required for this case. If a new vector is being created, the list length determines the maximum dimensions it is assigned in the data base. When an existing vector is overstored, the length of

the list may not exceed the dimensions previously assigned to that vector. To increase the dimensions of the vector, the vector must first be deleted and then re-created. When the list length is less than the length of the existing vector, excess elements are not updated.

Commands 8 through 11 show the syntax for updating an element, a column, a row, or an entire matrix. The use of command 11 requires that the matrix must have been previously initialized, so that its primary dimensions may be obtained from the data base. Command 8 is used to initialize a new matrix with the maximum dimensions given. As in the case of a vector, the maximum dimensions may not be exceeded. The matrix must be deleted and re-created to expand its size. Commands 9 and 10 show how to update a row and column of a matrix, respectively. If the data list is shorter than the matrix row or column previously initialized, the update starts at the beginning of the row or column and proceeds until the list is exhausted. The remaining elements are not updated. When commands 8 through 11 are used in the INQUIRY mode, any data list is ignored.

Command 12 in Figure 1 illustrates a data base feature that facilitates entry of tabular data. The I.D.s specify the headings of the table and the data are input as ID1 (1), ID2 (1), ID3 (1), ID1 (2), ID2 (2), and so on. The dimensions must match for all I.D.s, and the length of the data list must be a multiple of the number of I.D.s specified. The subscript information specified in parentheses is not required for vectors, but must be supplied for matrices.

Commands 13 and 14 in Figure 1 show how to delete a variable or variables. Once dimensions have been set for a variable, they are permanently associated with the I.D. and may not be changed unless the I.D. is first deleted by means of one of these commands.

The final command is numbered 15 in Figure 1. The END command is used at the end of the command string to trigger the closing of the files and data base and the orderly termination of the maintenance session.

A sample input data set for MAINT is shown in Table IV. The output print generated for this sample maintenance session is shown in Table V.

The sample session illustrates several important rules for the maintenance program. In command 3, note that I.D. 1 must be deleted before its maximum dimensions can be altered. Command 4 shows that a vector size need not be specified for initialization. Command 7 shows the input convention for elements of a matrix. The primary index varies fastest. Commands 11, 12, and 13 illustrate the flexibility of free-format input.

In the inquiry mode, note the action resulting from commands 17 and 18. Command 17 calls out a matrix I.D., but only one column of the matrix is returned if (*,*) is not present on the command line. Commands 20 through 26 recall the data stored above and 23 illustrates the output for multiple I.D. lists supplied in the INQUIRY mode. Commands 27 through 31 demonstrate the use of the DELETE command and the error message printed when a deleted variable is called.

TABLE IV. SAMPLE INPUT FOR MAINTENANCE PROGRAM

```
SET EXAMPLE    CREATE
ID 1 1981
ID 1            DELETE
ID 1 5 26. 1931
ID 1 (*) 7 4 1776
ID 2 (2,3) 1.
ID 2 (*,*) 11 21 12 22  13 23
ID 3 (4) 1.
ID 4 (4) 1.
ID 5·(4) 1.
ID 3 ID 4 ID 5 (*) 3 4     5       3       4 5
3 4 5
3 4 5
INQUIRY
RUN 122
ID 1
ID 2
ID 25
ID 25 (*,*)
SET EXAMPLE
ID 1
ID 2
ID 3 ID 4 ID 5
ID 3
ID 4
ID 5
UPDATE
ID 4 ID 5 DELETE
INQUIRY
ID 4
ID 5
END
```

Program 2--Bulk Data Listing (BLIST)

The purpose of the listing program, BLIST, is to provide a full or partial list of data base contents. The program provides for the listing of entire arrays or simply an index of variables stored for a given data set.

The BLIST software is comprised of a mainline, several "title" subroutines, and a generalized listing subroutine.

The mainline program first generates the random-access data bank from a sequential input tape file of the data base. It then proceeds to read and interpret the input data commands. The mainline processes the input commands sequentially and calls subroutines to generate the titles and variable lists specified. For each variable list, the code consists of a title subroutine call, followed by a listing subroutine call. When the END command is encountered, the mainline closes the random-access data bank. BLIST does not generate a new sequential tape file.

The "title" subroutines include subroutines GTITL and TITL1 through TITL10. GTITL retrieves the descriptive variable name corresponding to a specified GEOMETRY I.D. number. TITL1 through TITL10 contain the descriptive variable names for I.D. numbers of data sets of the type RUNXXXXX. If new variable I.D.s are added to the data base, additional title subroutines must be generated for BLIST.

TABLE V. SAMPLE OUTPUT FOR MAINTENANCE PROGRAM

```
0BANK LOAD COMPLETED
 ***SET EXAMPLE   CREATE
 ***ID 1 1981

 DATA SET EXAMPLE  ID NUMBER     1 SIZE (    1,    1)
   ELEMENT    OLD VALUE      NEW VALUE
 (    1,    1)   .0             1981.00
 ***ID 1           DELETE
 ***ID 1 5 26. 1981

 DATA SET EXAMPLE  ID NUMBER     1 SIZE (    3,    1)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   .0             5.00000      (    2,    1)   .0              26.0000
 (    3,    1)   .0             1981.00
 ***ID 1 (*) 7 4 1776

 DATA SET EXAMPLE  ID NUMBER     1 SIZE (    3,    1)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   5.00000        7.00000      (    2,    1)   26.0000         4.00000
 (    3,    1)   1981.00        1776.00
 ***ID 2 (2,3) 1.

 DATA SET EXAMPLE  ID NUMBER     2 SIZE (    2,    3)
   ELEMENT    OLD VALUE      NEW VALUE
 (    2,    3)   .0             1.00000
 ***ID 2 (*,*) 11 21 12 22  13 23

 DATA SET EXAMPLE  ID NUMBER     2 SIZE (    2,    3)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   .0             11.0000      (    2,    1)   26.0000         21.0000
 (    1,    2)   1981.00        12.0000      (    2,    2)   .404820E-78     22.0000
 (    1,    3)   .001581E-78    13.0000      (    2,    3)   1.00000         23.0000
 ***ID 3 (4) 1.

 DATA SET EXAMPLE  ID NUMBER     3 SIZE (    4,    1)
   ELEMENT    OLD VALUE      NEW VALUE
 (    4,    1)   .0             1.00000
 ***ID 4 (4) 1.

 DATA SET EXAMPLE  ID NUMBER     4 SIZE (    4,    1)
   ELEMENT    OLD VALUE      NEW VALUE
 (    4,    1)   .0             1.00000
 ***ID 5 (4) 1.

 DATA SET EXAMPLE  ID NUMBER     5 SIZE (    4,    1)
   ELEMENT    OLD VALUE      NEW VALUE
 (    4,    1)   .0             1.00000
 ***ID 3 ID 4 ID 5 (*) 3 4     5      3      4 5
 ***3 4 5
 ***3 4 5

 DATA SET EXAMPLE  ID NUMBER     3 SIZE (    4,    1)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   .0             3.00000      (    2,    1)   26.0000         3.00000
 (    3,    1)   1981.00        3.00000      (    4,    1)   1.00000         3.00000

 DATA SET EXAMPLE  ID NUMBER     4 SIZE (    4,    1)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   .0             4.00000      (    2,    1)   26.0000         4.00000
 (    3,    1)   1981.00        4.00000      (    4,    1)   1.00000         4.00000

 DATA SET EXAMPLE  ID NUMBER     5 SIZE (    4,    1)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   .0             5.00000      (    2,    1)   26.0000         5.00000
 (    3,    1)   1981.00        5.00000      (    4,    1)   1.00000         5.00000
 ***INQUIRY
 ***RUN 122
 ***ID 1

 DATA SET RUN  122 ID NUMBER     1 SIZE (    3,    1)
   ELEMENT    OLD VALUE      NEW VALUE          ELEMENT    OLD VALUE       NEW VALUE
 (    1,    1)   1.00000                      (    2,    1)   7.00000
 (    3,    1)   80.0000
 ***ID 2
```

TABLE V. (CONT)

```
DATA SET RUN  122 ID NUMBER     2 SIZE (    1,     1)
  ELEMENT    OLD VALUE       NEW VALUE
(   1,     1)    769.180
***ID 25

DATA SET RUN  122 ID NUMBER    25 SIZE (    5,     2)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   59.3718                       (    2,     1)   59.8754
(   3,     1)   48.2730                       (    4,     1)   59.9444
(   5,     1)   59.9812
***ID 25 (*,*)

DATA SET RUN  122 ID NUMBER    25 SIZE (    5,     2)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   59.3716                       (    2,     1)   59.8754
(   3,     1)   48.2730                       (    4,     1)   59.9444
(   5,     1)   59.9812
(   1,     2)   59.8329                       (    2,     2)   59.8120
(   3,     2)   59.8146                       (    4,     2)   59.8518
(   5,     2)   59.9251
***SET EXAMPLE
***ID 1

DATA SET EXAMPLE  ID NUMBER     1 SIZE (    3,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   7.00000                       (    2,     1)   4.00000
(   3,     1)   1776.00
***ID 2

DATA SET EXAMPLE  ID NUMBER     2 SIZE (    2,     3)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   11.0000                       (    2,     1)   21.0000
***ID 3 ID 4 ID 5

DATA SET EXAMPLE  ID NUMBER     3 SIZE (    4,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   3.00000                       (    2,     1)   3.00000
(   3,     1)   3.00000                       (    4,     1)   3.00000
DATA SET EXAMPLE  ID NUMBER     4 SIZE (    4,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   4.00000                       (    2,     1)   4.00000
(   3,     1)   4.00000                       (    4,     1)   4.00000
DATA SET EXAMPLE  ID NUMBER     5 SIZE (    4,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   5.00000                       (    2,     1)   5.00000
(   3,     1)   5.00000                       (    4,     1)   5.00000
***ID 3

DATA SET EXAMPLE  ID NUMBER     3 SIZE (    4,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   3.00000                       (    2,     1)   3.00000
(   3,     1)   3.00000                       (    4,     1)   3.00000
***ID 4

DATA SET EXAMPLE  ID NUMBER     4 SIZE (    4,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   4.00000                       (    2,     1)   4.00000
(   3,     1)   4.00000                       (    4,     1)   4.00000
***ID 5

DATA SET EXAMPLE  ID NUMBER     5 SIZE (    4,     1)
  ELEMENT    OLD VALUE       NEW VALUE            ELEMENT     OLD VALUE      NEW VALUE
(   1,     1)   5.00000                       (    2,     1)   5.00000
(   3,     1)   5.00000                       (    4,     1)   5.00000
***UPDATE
***ID 4 ID 5 DELETE
***INQUIRY
***ID 4
OUNABLE TO LOCATE     4 IN SET EXAMPLE
 COMMAND ERROR
***ID 5
OUNABLE TO LOCATE     5 IN SET EXAMPLE
 COMMAND ERROR

***END

END OF UPDATES
```

The general listing subroutine, WRIT, prints the variable description obtained from the title subroutine and then retrieves and prints the data arrays if the program is in the ARRAY mode. If the program is in the INDEX mode, only the title line is printed. Subroutine WRIT is general and may be used to list new variables added to the data base, providing the title information is correctly supplied.

Input to BLIST consists of a series of commands using key words. The commands may be arranged to list all or any subset of the data base. The commands for BLIST are shown in Figure 3. Each command line is 80 characters long. Key words are underlined, and when shown as beginning in column one, they must appear there. When blanks are underlined, they are an integral part of the key word.

The commands shown in Figure 5 are of three major types. The mode command specifies the function to be performed by the program, and the set command specifies the data set or sets to be listed with optional I.D. specification. The end command signals the end of the run.

```
1 INDEX
2 ARRAY
3 ALL
4 RUN ###
5 GEOMETRY
6 RUN ###    [ID ##]
7 GEOMETRY   [ID ##]
8 END
```

TE81-5063

Figure 3. Listing program commands.

The mode command, INDEX, in Figure 3 will cause the program to list an index of variable names stored for the following data set names. In the ARRAY mode, variable names and their associated data arrays are listed. The program begins in the ARRAY mode and remains in that mode until another mode command is encountered. The program mode may be changed at any point in a listing series of commands.

The second major type of command is the set specification command, shown on lines 3, 4, and 5 of Figure 3. The ALL command will cause all data sets RUNXXXXX, to be listed according to the last mode specified. The RUN command will produce a list of all data associated with the specified run number. Geometry data, or an index of geometry data, will be listed if command 5 is present in the series of commands. Shown in brackets in commands 4 and 5 is the optional I.D. specification. If a list of one data array is desired, adding the I.D. number to the set specification command will produce the single array list. The I.D. option provides the flexibility for listing any subset of the data base.

Finally, the END command is used to terminate the listing session. The command triggers the orderly closing of files.

A sample input data set for BLIST is shown in Table VI. The resulting output listing appears in Table VII. Note the results of mode changes and the use of command 4 to generate special lists.

TABLE VI. SAMPLE INPUT FOR LISTING PROGRAM.

```
INDEX
ALL
GEOMETRY
ARRAY
RUN 96
RUN 122  ID 3
RUN 122 ID 5
GEOMETRY ID 5
END
```

3.1.2 Interpolation Functions

Two interpolation function subroutines are provided to aid the analyst. The function ANTRP interpolates data associated with endwall instrumentation, and BNTRP interpolates data associated with the traversing exit probe. ANTRP and BNTRP interact with the data base, retrieving data as needed, and both use subroutine GNTRP to obtain an interpolated value. Subroutine GNTRP is a user-transparent routine that performs a weighted least-squares fit of the data and returns the single value for given coordinates.

Subroutines FCTNL, CHOLEP, and CBSBMP are matrix routines used by GNTRP to determine the curve-fit coefficients. They are also user-transparent.

TABLE VII. SAMPLE OUTPUT FOR LISTING PROGRAM

```
OBANK LOAD COMPLETED
                                                                                      ***
***INDEX

                                                                                      ***
***ALL

DATA SET RUN   91 ID NUMBER    1 SIZE (   3,    1)      DATE: MO,DAY,YEAR
DATA SET RUN   91 ID NUMBER   73 SIZE (   1,    1)      GAS TEMPERATURE DURING ENDWALL READING
DATA SET RUN   91 ID NUMBER   74 SIZE (  53,    1)      H.T. ENDWALL HOT SIDE TEMPERATURE
DATA SET RUN   91 ID NUMBER   75 SIZE (  31,    1)      H.T. ENDWALL COLD SIDE TEMPERATURE
DATA SET RUN   91 ID NUMBER   76 SIZE (   1,    1)      GAS TEMPERATURE DURING VANE T/C READING
DATA SET RUN   91 ID NUMBER   77 SIZE ( 113,    1)      VANE TEMPERATURE
DATA SET RUN   91 ID NUMBER   78 SIZE (  90,    1)      ADIABATIC ENDWALL TEMPERATURE
DATA SET RUN   91 ID NUMBER   79 SIZE (   1,    1)      INSULATION TEMPERATURE 1
DATA SET RUN   91 ID NUMBER   80 SIZE (   1,    1)      INSULATION TEMPERATURE 2
DATA SET RUN   91 ID NUMBER   81 SIZE (   1,    1)      GAGE BLOCK TEMPERATURE 1
DATA SET RUN   91 ID NUMBER   82 SIZE (   1,    1)      GAGE BLOCK TEMPERATURE 2
DATA SET RUN   91 ID NUMBER  157 SIZE ( 338,    1)      H.T. ENDWALL NODE TEMPERATURE
DATA SET RUN   91 ID NUMBER  158 SIZE ( 338,    1)      H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2
DATA SET RUN   91 ID NUMBER  159 SIZE ( 338,    1)      H.T. ENDWALL NODE STANTON NUMBER

DATA SET RUN   93 ID NUMBER    1 SIZE (   3,    1)      DATE: MO,DAY,YEA
DATA SET RUN   93 ID NUMBER   73 SIZE (   1,    1)      GAS TEMPERATURE DURING ENDWALL READING
DATA SET RUN   93 ID NUMBER   74 SIZE (  53,    1)      H.T. ENDWALL HOT SIDE TEMPERATURE
DATA SET RUN   93 ID NUMBER   75 SIZE (  31,    1)      H.T. ENDWALL COLD SIDE TEMPERATURE
DATA SET RUN   93 ID NUMBER   76 SIZE (   1,    1)      GAS TEMPERATURE DURING VANE T/C READING
DATA SET RUN   93 ID NUMBER   77 SIZE ( 113,    1)      VANE TEMPERATURE
DATA SET RUN   93 ID NUMBER   78 SIZE (  90,    1)      ADIABATIC ENDWALL TEMPERATURE
DATA SET RUN   93 ID NUMBER   79 SIZE (   1,    1)      INSULATION TEMPERATURE 1
DATA SET RUN   93 ID NUMBER   80 SIZE (   1,    1)      INSULATION TEMPERATURE 2
DATA SET RUN   93 ID NUMBER   81 SIZE (   1,    1)      GAGE BLOCK TEMPERATURE 1
DATA SET RUN   93 ID NUMBER   82 SIZE (   1,    1)      GAGE BLOCK TEMPERATURE 2
DATA SET RUN   93 ID NUMBER  157 SIZE ( 338,    1)      H.T. ENDWALL NODE TEMPERATURE
DATA SET RUN   93 ID NUMBER  158 SIZE ( 338,    1)      H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2
DATA SET RUN   93 ID NUMBER  159 SIZE ( 338,    1)      H.T. ENDWALL NODE STANTON NUMBER

DATA SET RUN   94 ID NUMBER    1 SIZE (   3,    1)      DATE: MO,DAY,YEAR
DATA SET RUN   94 ID NUMBER   73 SIZE (   1,    1)      GAS TEMPERATURE DURING ENDWALL READING
DATA SET RUN   94 ID NUMBER   74 SIZE (  53,    1)      H.T. ENDWALL HOT SIDE TEMPERATURE
DATA SET RUN   94 ID NUMBER   75 SIZE (  31,    1)      H.T. ENDWALL COLD SIDE TEMPERATURE
DATA SET RUN   94 ID NUMBER   76 SIZE (   1,    1)      GAS TEMPERATURE DURING VANE T/C READING
DATA SET RUN   94 ID NUMBER   77 SIZE ( 113,    1)      VANE TEMPERATURE
DATA SET RUN   94 ID NUMBER   78 SIZE (  90,    1)      ADIABATIC ENDWALL TEMPERATURE
DATA SET RUN   94 ID NUMBER   79 SIZE (   1,    1)      INSULATION TEMPERATURE 1
DATA SET RUN   94 ID NUMBER   80 SIZE (   1,    1)      INSULATION TEMPERATURE 2
DATA SET RUN   94 ID NUMBER   81 SIZE (   1,    1)      GAGE BLOCK TEMPERATURE 1
DATA SET RUN   94 ID NUMBER   82 SIZE (   1,    1)      GAGE BLOCK TEMPERATURE 2


DATA SET RUN   96 ID NUMBER  157 SIZE ( 338,    1)      H.T. ENDWALL NODE TEMPERATURE
DATA SET RUN   96 ID NUMBER  158 SIZE ( 338,    1)      H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2
DATA SET RUN   96 ID NUMBER  159 SIZE ( 338,    1)      H.T. ENDWALL NODE STANTON NUMBER
```

TABLE VII. (CONT)

DATA SET RUN	116	ID NUMBER 157	SIZE (338, 1)	H.T. ENDWALL NODE TEMPERATURE
DATA SET RUN	116	ID NUMBER 158	SIZE (338, 1)	H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2
DATA SET RUN	116	ID NUMBER 159	SIZE (338, 1)	H.T. ENDWALL NODE STANTON NUMBER
DATA SET RUN	122	ID NUMBER 1	SIZE (3, 1)	DATE: MO,DAY,YEAR
DATA SET RUN	122	ID NUMBER 2	SIZE (1, 1)	INLET TOTAL TEMPERATURE
DATA SET RUN	122	ID NUMBER 3	SIZE (1, 1)	IDEAL EXIT MACH NUMBER
DATA SET RUN	122	ID NUMBER 4	SIZE (1, 1)	INLET TOTAL PRESSURE
DATA SET RUN	122	ID NUMBER 5	SIZE (1, 1)	AVERAGE INLET STATIC PRESSURE
DATA SET RUN	122	ID NUMBER 6	SIZE (1, 1)	AVERAGE EXIT STATIC PRESSURE
DATA SET RUN	122	ID NUMBER 7	SIZE (1, 1)	INLET REYNOLDS NUMBER/ 10E6 REF. CHORD
DATA SET RUN	122	ID NUMBER 8	SIZE (1, 1)	IDEAL EXIT REYNOLDS NUMBER/ 10E6
DATA SET RUN	122	ID NUMBER 12	SIZE (1, 1)	INLET MACH NUMBER
DATA SET RUN	122	ID NUMBER 13	SIZE (1, 1)	EXPANSION RATIO PT1/PSX
DATA SET RUN	122	ID NUMBER 14	SIZE (1, 1)	PRESSURE RATIO PSX/PS1
DATA SET RUN	122	ID NUMBER 15	SIZE (1, 1)	INLET V/VCRITICAL
DATA SET RUN	122	ID NUMBER 16	SIZE (1, 1)	EXIT V/VCRITICAL
DATA SET RUN	122	ID NUMBER 25	SIZE (5, 2)	CORE RAKE PRESSURES
DATA SET RUN	122	ID NUMBER 26	SIZE (5, 2)	CORE RAKE TEMPERATURES
DATA SET RUN	122	ID NUMBER 27	SIZE (9, 1)	SIDEWALL STATIC PRESSURES -INLET
DATA SET RUN	122	ID NUMBER 28	SIZE (9, 1)	SIDEWALL STATIC PRESSURE -EXIT ROW1
DATA SET RUN	122	ID NUMBER 29	SIZE (13, 1)	SIDEWALL STATIC PRESSURE -EXIT ROW2
DATA SET RUN	122	ID NUMBER 30	SIZE (16, 2)	BOUNDARY LAYER RAKE PRESSURE
DATA SET RUN	122	ID NUMBER 31	SIZE (16, 1)	BOUNDARY LAYER RAKE TEMPERATURE
DATA SET RUN	122	ID NUMBER 32	SIZE (42, 1)	ENDWALL STATIC PRESSURE
DATA SET RUN	122	ID NUMBER 33	SIZE (36, 1)	VANE STATIC PRESSURE -PRESSURE SIDE
DATA SET RUN	122	ID NUMBER 34	SIZE (30, 1)	VANE STATIC PRESSURE -SUCTION SIDE
DATA SET RUN	122	ID NUMBER 37	SIZE (1, 1)	INLET TRAVERSE PROBE LENGTH
DATA SET RUN	122	ID NUMBER 38	SIZE (1, 1)	INLET TRAVERSE PROBE WIDTH
DATA SET RUN	122	ID NUMBER 39	SIZE (1, 1)	INLET TRAVERSE EXTERIOR ANGLE
DATA SET RUN	122	ID NUMBER 40	SIZE (36, 1)	INLET TRAVERSE DISTANCE FROM WALL
DATA SET RUN	122	ID NUMBER 41	SIZE (36, 1)	INLET TRAVERSE CORRECTED PRESSURE
DATA SET RUN	122	ID NUMBER 42	SIZE (36, 1)	INLET TRAVERSE TEMPERATURE
DATA SET RUN	122	ID NUMBER 60	SIZE (9, 1)	EXIT PROBE SPANWISE POSITION
DATA SET RUN	122	ID NUMBER 61	SIZE (25, 9)	EXIT PROBE VERTICAL POSITION
DATA SET RUN	122	ID NUMBER 62	SIZE (25, 9)	EXIT PROBE REFERENCE PRESSURE
DATA SET RUN	122	ID NUMBER 63	SIZE (25, 9)	EXIT PROBE TOTAL PRESSURE
DATA SET RUN	122	ID NUMBER 64	SIZE (25, 9)	EXIT PROBE STATIC PRESSURE 1
DATA SET RUN	122	ID NUMBER 65	SIZE (25, 9)	EXIT PROBE STATIC PRESSURE 2
DATA SET RUN	122	ID NUMBER 66	SIZE (25, 9)	EXIT PROBE STATIC PRESSURE 3
DATA SET RUN	122	ID NUMBER 67	SIZE (25, 9)	EXIT PROBE STATIC PRESSURE 4
DATA SET RUN	122	ID NUMBER 68	SIZE (25, 9)	EXIT PROBE TOTAL TEMPERATURE
DATA SET RUN	122	ID NUMBER 69	SIZE (1, 1)	EXIT PROBE ANGLE
DATA SET RUN	122	ID NUMBER 73	SIZE (1, 1)	GAS TEMPERATURE DURING ENDWALL READING
DATA SET RUN	122	ID NUMBER 74	SIZE (53, 1)	H.T. ENDWALL HOT SIDE TEMPERATURE
DATA SET RUN	122	ID NUMBER 75	SIZE (31, 1)	H.T. ENDWALL COLD SIDE TEMPERATURE
DATA SET RUN	122	ID NUMBER 76	SIZE (1, 1)	GAS TEMPERATURE DURING VANE T/C READING
DATA SET RUN	122	ID NUMBER 77	SIZE (113, 1)	VANE TEMPERATURE
DATA SET RUN	122	ID NUMBER 78	SIZE (90, 1)	ADIABATIC ENDWALL TEMPERATURE
DATA SET RUN	122	ID NUMBER 79	SIZE (1, 1)	INSULATION TEMPERATURE 1
DATA SET RUN	122	ID NUMBER 80	SIZE (1, 1)	INSULATION TEMPERATURE 2
DATA SET RUN	122	ID NUMBER 81	SIZE (1, 1)	GAGE BLOCK TEMPERATURE 1
DATA SET RUN	122	ID NUMBER 82	SIZE (1, 1)	GAGE BLOCK TEMPERATURE 2
DATA SET RUN	122	ID NUMBER 85	SIZE (36, 1)	INLET TRAVERSE MACH NUMBER
DATA SET RUN	122	ID NUMBER 86	SIZE (36, 1)	INLET TRAVERSE VELOCITY
DATA SET RUN	122	ID NUMBER 90	SIZE (25, 9)	EXIT TOTAL PRESSURE
DATA SET RUN	122	ID NUMBER 91	SIZE (25, 9)	EXIT STATIC PRESSURE
DATA SET RUN	122	ID NUMBER 92	SIZE (25, 9)	EXIT MACH NUMBER
DATA SET RUN	122	ID NUMBER 93	SIZE (25, 9)	EXIT YAW ANGLE

TABLE VII. (CONT)

DATA SET	RUN 122	ID NUMBER 94	SIZE	(25, 9)	EXIT PITCH ANGLE
DATA SET	RUN 122	ID NUMBER 95	SIZE	(25, 9)	EXIT %P (PASSAGE LOCATION)
DATA SET	RUN 122	ID NUMBER 96	SIZE	(25, 9)	EXIT V/V*
DATA SET	RUN 122	ID NUMBER 97	SIZE	(25, 9)	EXIT V/V* BASED ON IDEAL PTX,PSPROBE
DATA SET	RUN 122	ID NUMBER 98	SIZE	(25, 9)	EXIT BETA2
DATA SET	RUN 122	ID NUMBER 99	SIZE	(25, 9)	EXIT OMEGA
DATA SET	RUN 122	ID NUMBER 100	SIZE	(25, 9)	EXIT EBAR
DATA SET	RUN 122	ID NUMBER 101	SIZE	(25, 9)	EXIT DS/CP
DATA SET	RUN 122	ID NUMBER 102	SIZE	(25, 9)	EXIT MASS FLUX
DATA SET	RUN 122	ID NUMBER 103	SIZE	(25, 9)	EXIT VELOCITY
DATA SET	RUN 122	ID NUMBER 104	SIZE	(25, 9)	EXIT AXIAL VELOCITY
DATA SET	RUN 122	ID NUMBER 105	SIZE	(25, 9)	EXIT TANGENTIAL VELOCITY
DATA SET	RUN 122	ID NUMBER 106	SIZE	(25, 9)	EXIT SPANWISE VELOCITY
DATA SET	RUN 122	ID NUMBER 107	SIZE	(25, 9)	EXIT TOTAL TEMPERATURE AT CONE PROBE
DATA SET	RUN 122	ID NUMBER 109	SIZE	(9, 1)	EXIT AVERAGED MASS FLOW
DATA SET	RUN 122	ID NUMBER 110	SIZE	(9, 1)	EXIT MASS-AVERAGED TOTAL PRESSURE
DATA SET	RUN 122	ID NUMBER 111	SIZE	(9, 1)	EXIT MASS-AVERAGED STATIC PRESSURE
DATA SET	RUN 122	ID NUMBER 112	SIZE	(9, 1)	EXIT MASS-AVERAGED MACH NUMBER
DATA SET	RUN 122	ID NUMBER 113	SIZE	(9, 1)	EXIT MASS-AVERAGED YAW ANGLE
DATA SET	RUN 122	ID NUMBER 114	SIZE	(9, 1)	EXIT MASS-AVERAGED PITCH ANGLE
DATA SET	RUN 122	ID NUMBER 115	SIZE	(9, 1)	EXIT MASS-AVERAGED PT/PS
DATA SET	RUN 122	ID NUMBER 116	SIZE	(9, 1)	EXIT MASS-AVERAGED V/VCRITICAL
DATA SET	RUN 122	ID NUMBER 117	SIZE	(9, 1)	EXIT MASS-AVERAGED V/V* FROM PTX,PSPROBE
DATA SET	RUN 122	ID NUMBER 118	SIZE	(9, 1)	EXIT MASS-AVERAGED BETA2
DATA SET	RUN 122	ID NUMBER 119	SIZE	(9, 1)	EXIT MASS-AVERAGED (PTX-PT)
DATA SET	RUN 122	ID NUMBER 120	SIZE	(9, 1)	EXIT MASS-AVERAGED (PTX-PT)/PTX
DATA SET	RUN 122	ID NUMBER 121	SIZE	(9, 1)	EXIT MASS-AVERAGED OMEGA
DATA SET	RUN 122	ID NUMBER 122	SIZE	(9, 1)	EXIT MASS-AVERAGED EBAR
DATA SET	RUN 122	ID NUMBER 123	SIZE	(9, 1)	EXIT MASS-AVERAGED DS/CP
DATA SET	RUN 122	ID NUMBER 124	SIZE	(9, 1)	EXIT MASS-AVERAGED PS2/PS1
DATA SET	RUN 122	ID NUMBER 125	SIZE	(9, 1)	EXIT MASS-AVERAGED V2
DATA SET	RUN 122	ID NUMBER 126	SIZE	(9, 1)	EXIT MASS-AVERAGED V2AXIAL
DATA SET	RUN 122	ID NUMBER 127	SIZE	(9, 1)	EXIT MASS-AVERAGED V2TANGENTIAL
DATA SET	RUN 122	ID NUMBER 128	SIZE	(9, 1)	EXIT MASS-AVERAGED V2SPANWISE
DATA SET	RUN 122	ID NUMBER 129	SIZE	(9, 1)	EXIT MASS-AVERAGED T2 AT CONE PROBE
DATA SET	RUN 122	ID NUMBER 130	SIZE	(9, 1)	EXIT MIXED OUT PT3
DATA SET	RUN 122	ID NUMBER 131	SIZE	(9, 1)	EXIT MIXED OUT PS3
DATA SET	RUN 122	ID NUMBER 132	SIZE	(9, 1)	EXIT MIXED OUT EXPANSION RATIO
DATA SET	RUN 122	ID NUMBER 133	SIZE	(9, 1)	EXIT MIXED OUT PS2/PS1
DATA SET	RUN 122	ID NUMBER 134	SIZE	(9, 1)	EXIT MIXED OUT MACH NUMBER
DATA SET	RUN 122	ID NUMBER 135	SIZE	(9, 1)	EXIT MIXED OUT BETA3
DATA SET	RUN 122	ID NUMBER 136	SIZE	(9, 1)	EXIT MIXED OUT OMEGA3
DATA SET	RUN 122	ID NUMBER 137	SIZE	(9, 1)	EXIT MIXED OUT EBAR3
DATA SET	RUN 122	ID NUMBER 138	SIZE	(9, 1)	EXIT MIXED OUT DS/CP
DATA SET	RUN 122	ID NUMBER 139	SIZE	(9, 1)	EXIT MIXED OUT V/V*3
DATA SET	RUN 122	ID NUMBER 140	SIZE	(9, 1)	EXIT MIXED OUT V/V* BASED ON PTX, PS3
DATA SET	RUN 122	ID NUMBER 141	SIZE	(9, 1)	EXIT MIXED OUT V3
DATA SET	RUN 122	ID NUMBER 142	SIZE	(9, 1)	EXIT MIXED OUT V3 AXIAL
DATA SET	RUN 122	ID NUMBER 143	SIZE	(9, 1)	EXIT MIXED OUT V3 TANGENTIAL
DATA SET	RUN 122	ID NUMBER 145	SIZE	(1, 1)	EXIT AVERAGE MASS FLUX
DATA SET	RUN 122	ID NUMBER 146	SIZE	(1, 1)	EXIT AVERAGE PT3
DATA SET	RUN 122	ID NUMBER 147	SIZE	(1, 1)	EXIT AVERAGE TT3
DATA SET	RUN 122	ID NUMBER 148	SIZE	(1, 1)	EXIT AVERAGE MACH NUMBER3
DATA SET	RUN 122	ID NUMBER 149	SIZE	(1, 1)	EXIT AVERAGE BETA3
DATA SET	RUN 122	ID NUMBER 150	SIZE	(1, 1)	EXIT AVERAGE OMEGA3
DATA SET	RUN 122	ID NUMBER 151	SIZE	(1, 1)	EXIT AVERAGE EBAR3
DATA SET	RUN 122	ID NUMBER 157	SIZE	(338, 1)	H.T. ENDWALL NODE TEMPERATURE
DATA SET	RUN 122	ID NUMBER 158	SIZE	(338, 1)	H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2
DATA SET	RUN 122	ID NUMBER 159	SIZE	(338, 1)	H.T. ENDWALL NODE STANTON NUMBER

TABLE VII. (CONT)

```
***GEOMETRY                                                                                ***
DATA SET GEOMETRY ID NUMBER    1 SIZE (   2,  53)     H.T. ENDWALL HOT SURFACE T/C COORDINATES
DATA SET GEOMETRY ID NUMBER    2 SIZE (   2,  31)     H.T.ENDWALL COLD SURFACE T/C COORDINATES
DATA SET GEOMETRY ID NUMBER    3 SIZE (   2, 338)     H.T. ENDWALL NODE COORDINATES
DATA SET GEOMETRY ID NUMBER    4 SIZE (   2,  90)     ADIABATIC ENDWALL T/C COORDINATES
DATA SET GEOMETRY ID NUMBER    5 SIZE (   2,  42)     ENDWALL STATIC PRESSURE TAP COORDINATES
DATA SET GEOMETRY ID NUMBER    6 SIZE (   2, 113)     VANE T/C COORDINATES
DATA SET GEOMETRY ID NUMBER    7 SIZE (   3,  85)     H.T. ENDWALL SURFACE BOUNDARY POINTS

***ARRAY                                                                                   ***

***RUN 96                                                                                  ***
```

DATA SET RUN 96 ID NUMBER 157 SIZE (338, 1) H.T. ENDWALL NODE TEMPERATURE

ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE
(1, 1)	456.175	(2, 1)	456.669	(3, 1)	457.364	(4, 1)	458.245
(5, 1)	458.930	(6, 1)	460.696	(7, 1)	461.198	(8, 1)	461.401
(9, 1)	461.203	(10, 1)	460.655	(11, 1)	459.480	(12, 1)	458.463
(13, 1)	459.539	(14, 1)	460.179	(15, 1)	459.558	(16, 1)	457.496
(17, 1)	455.582	(18, 1)	453.778	(19, 1)	455.526	(20, 1)	458.407
(21, 1)	457.315	(22, 1)	454.370	(23, 1)	452.458	(24, 1)	451.866
(25, 1)	451.297	(26, 1)	450.636	(27, 1)	451.281	(28, 1)	452.615
(29, 1)	452.497	(30, 1)	454.036	(31, 1)	456.076	(32, 1)	455.135
(33, 1)	454.212	(34, 1)	453.344	(35, 1)	451.518	(36, 1)	450.592
(37, 1)	456.634	(38, 1)	456.944	(39, 1)	457.232	(40, 1)	456.569
(41, 1)	455.924	(42, 1)	455.310	(43, 1)	454.700	(44, 1)	454.037
(45, 1)	457.637	(46, 1)	458.062	(47, 1)	458.524	(48, 1)	457.918
(49, 1)	457.324	(50, 1)	456.739	(51, 1)	456.283	(52, 1)	455.811
(53, 1)	457.390	(54, 1)	457.011	(55, 1)	456.374	(56, 1)	456.239
(57, 1)	456.209	(58, 1)	456.273	(59, 1)	456.078	(60, 1)	456.330
(61, 1)	455.567	(62, 1)	454.753	(63, 1)	453.903	(64, 1)	454.297
(65, 1)	454.771	(66, 1)	452.880	(67, 1)	451.900	(68, 1)	450.926
(69, 1)	451.748	(70, 1)	452.540	(71, 1)	448.550	(72, 1)	447.072
(73, 1)	446.360	(74, 1)	447.801	(75, 1)	449.372	(76, 1)	446.355
(77, 1)	446.625	(78, 1)	447.056	(79, 1)	449.261	(80, 1)	451.344
(81, 1)	446.791	(82, 1)	446.339	(83, 1)	446.282	(84, 1)	448.993
(85, 1)	454.490	(86, 1)	445.502	(87, 1)	444.168	(88, 1)	443.284
(89, 1)	443.4[illegible]3	(90, 1)	443.431	(91, 1)	443.552	(92, 1)	444.874
(93, 1)	445.630	(94, 1)	445.169	(95, 1)	444.140	(96, 1)	443.419
(97, 1)	443.240	(98, 1)	443.202	(99, 1)	450.385	(100, 1)	449.667
(101, 1)	448.902	(102, 1)	446.546	(103, 1)	444.528	(104, 1)	453.826
(105, 1)	453.707	(106, 1)	453.566	(107, 1)	452.376	(108, 1)	450.775
(109, 1)	455.421	(110, 1)	454.687	(111, 1)	447.229	(112, 1)	445.911
(113, 1)	444.790	(114, 1)	444.169	(115, 1)	443.488	(116, 1)	443.215
(117, 1)	443.067	(118, 1)	443.165	(119, 1)	443.229	(120, 1)	443.351
(121, 1)	443.548	(122, 1)	443.060	(123, 1)	442.954	(124, 1)	443.780
(125, 1)	443.602	(126, 1)	443.566	(127, 1)	443.070	(128, 1)	443.042
(129, 1)	444.880	(130, 1)	446.286	(131, 1)	448.456	(132, 1)	449.665
(133, 1)	450.368	(134, 1)	450.804	(135, 1)	446.561	(136, 1)	444.711
(137, 1)	443.663	(138, 1)	444.977	(139, 1)	447.162	(140, 1)	447.602
(141, 1)	448.006	(142, 1)	443.876	(143, 1)	444.120	(144, 1)	445.321
(145, 1)	445.653	(146, 1)	446.548	(147, 1)	448.072	(148, 1)	451.340
(149, 1)	454.551	([illegible], 1)	454.918	(151, 1)	455.778	(152, 1)	456.979
(153, 1)	453.53[illegible]	([illegible], 1)	450.025	(155, 1)	458.074	(156, 1)	459.156
(157, 1)	460.229	([illegible], 1)	456.015	(159, 1)	451.947	(160, 1)	461.464
(161, 1)	462.840	(162, 1)	463.323	(163, 1)	458.570	(164, 1)	454.210
(165, 1)	462.566	(166, 1)	464.982	(167, 1)	467.447	(168, 1)	468.102
(169, 1)	468.262	(170, 1)	469.086	(171, 1)	467.822	(172, 1)	466.039
(173, 1)	[illegible]	(174, 1)	460.680	(175, 1)	462.623	(176, 1)	464.398
(177, 1)	[illegible]	([illegible], 1)	456.386	(179, 1)	458.012	(180, 1)	459.[illegible]11

TABLE VII. (CONT)

ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE
(181, 1)	459.770	(182, 1)	460.790	(183, 1)	460.819	(184, 1)	462.114
(185, 1)	463.821	(186, 1)	464.417	(187, 1)	465.701	(188, 1)	468.063
(189, 1)	466.436	(190, 1)	464.611	(191, 1)	470.371	(192, 1)	472.625
(193, 1)	474.573	(194, 1)	472.494	(195, 1)	470.016	(196, 1)	475.437
(197, 1)	475.671	(198, 1)	475.872	(199, 1)	473.662	(200, 1)	471.295
(201, 1)	476.238	(202, 1)	475.961	(203, 1)	475.213	(204, 1)	473.686
(205, 1)	471.771	(206, 1)	470.125	(207, 1)	469.839	(208, 1)	469.579
(209, 1)	473.548	(210, 1)	471.336	(211, 1)	468.681	(212, 1)	470.153
(213, 1)	470.374	(214, 1)	470.201	(215, 1)	470.350	(216, 1)	470.349
(217, 1)	466.754	(218, 1)	464.964	(219, 1)	463.433	(220, 1)	466.291
(221, 1)	467.347	(222, 1)	468.419	(223, 1)	469.165	(224, 1)	469.802
(225, 1)	477.152	(226, 1)	478.844	(227, 1)	480.383	(228, 1)	479.935
(229, 1)	479.445	(230, 1)	478.915	(231, 1)	475.975	(232, 1)	472.553
(233, 1)	478.427	(234, 1)	480.357	(235, 1)	482.166	(236, 1)	481.811
(237, 1)	481.212	(238, 1)	476.414	(239, 1)	478.352	(240, 1)	480.495
(241, 1)	481.610	(242, 1)	482.159	(243, 1)	470.769	(244, 1)	473.225
(245, 1)	475.802	(246, 1)	477.690	(247, 1)	479.260	(248, 1)	466.833
(249, 1)	469.447	(250, 1)	471.829	(251, 1)	472.457	(252, 1)	473.558
(253, 1)	475.502	(254, 1)	478.984	(255, 1)	482.355	(256, 1)	482.447
(257, 1)	482.841	(258, 1)	483.340	(259, 1)	480.544	(260, 1)	478.001
(261, 1)	485.261	(262, 1)	486.801	(263, 1)	487.925	(264, 1)	485.132
(265, 1)	482.706	(266, 1)	512.243	(267, 1)	487.206	(268, 1)	486.397
(269, 1)	485.537	(270, 1)	484.375	(271, 1)	483.292	(272, 1)	485.594
(273, 1)	485.326	(274, 1)	484.758	(275, 1)	483.348	(276, 1)	481.977
(277, 1)	485.130	(278, 1)	485.519	(279, 1)	485.954	(280, 1)	483.422
(281, 1)	481.365	(282, 1)	490.300	(283, 1)	492.770	(284, 1)	495.382
(285, 1)	495.578	(286, 1)	495.110	(287, 1)	493.988	(288, 1)	491.081
(289, 1)	488.393	(290, 1)	486.378	(291, 1)	489.444	(292, 1)	492.549
(293, 1)	493.837	(294, 1)	494.787	(295, 1)	485.925	(296, 1)	489.525
(297, 1)	492.786	(298, 1)	493.133	(299, 1)	493.376	(300, 1)	495.766
(301, 1)	498.662	(302, 1)	501.452	(303, 1)	502.310	(304, 1)	502.624
(305, 1)	501.711	(306, 1)	498.597	(307, 1)	495.551	(308, 1)	503.103
(309, 1)	504.271	(310, 1)	505.208	(311, 1)	501.563	(312, 1)	498.312
(313, 1)	505.796	(314, 1)	506.244	(315, 1)	506.550	(316, 1)	506.829
(317, 1)	508.354	(318, 1)	509.759	(319, 1)	511.946	(320, 1)	514.162
(321, 1)	516.393	(322, 1)	513.539	(323, 1)	510.234	(324, 1)	502.647
(325, 1)	503.862	(326, 1)	507.458	(327, 1)	504.578	(328, 1)	508.868
(329, 1)	511.956	(330, 1)	511.492	(331, 1)	509.747	(332, 1)	512.072
(333, 1)	506.256	(334, 1)	508.104	(335, 1)	501.835	(336, 1)	497.637
(337, 1)	488.685	(338, 1)	487.976				

DATA SET RUN 96 ID NUMBER 158 SIZE (338, 1) H.T. ENDWALL NODE HEAT FLUX - BTU/HR/IN2

ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE
(1, 1)	-137.648	(2, 1)	-143.304	(3, 1)	-149.129	(4, 1)	-155.045
(5, 1)	-165.119	(6, 1)	-184.544	(7, 1)	-202.615	(8, 1)	-199.061
(9, 1)	-196.102	(10, 1)	-192.202	(11, 1)	-174.290	(12, 1)	-154.782
(13, 1)	-162.596	(14, 1)	-166.439	(15, 1)	-163.560	(16, 1)	-150.723
(17, 1)	-141.299	(18, 1)	-136.764	(19, 1)	-157.935	(20, 1)	-181.618
(21, 1)	-151.060	(22, 1)	-134.868	(23, 1)	-124.760	(24, 1)	-120.218
(25, 1)	-116.331	(26, 1)	-112.990	(27, 1)	-119.519	(28, 1)	-129.032
(29, 1)	-123.677	(30, 1)	-127.931	(31, 1)	-134.559	(32, 1)	-130.005
(33, 1)	-125.766	(34, 1)	-122.048	(35, 1)	-116.614	(36, 1)	-112.706
(37, 1)	-136.480	(38, 1)	-137.580	(39, 1)	-138.616	(40, 1)	-136.472
(41, 1)	-134.473	(42, 1)	-132.680	(43, 1)	-129.284	(44, 1)	-125.731
(45, 1)	-140.074	(46, 1)	-141.569	(47, 1)	-143.171	(48, 1)	-141.605
(49, 1)	-140.110	(50, 1)	-138.672	(51, 1)	-136.723	(52, 1)	-134.737
(53, 1)	-147.952	(54, 1)	-154.006	(55, 1)	-159.721	(56, 1)	-156.049
(57, 1)	-152.668	(58, 1)	-149.601	(59, 1)	-145.364	(60, 1)	-141.861
(61, 1)	-144.169	(62, 1)	-138.432	(63, 1)	-132.641	(64, 1)	-132.431
(65, 1)	-132.346	(66, 1)	-126.936	(67, 1)	-121.817	(68, 1)	-116.727
(69, 1)	-117.530	(70, 1)	-119.384	(71, 1)	-110.852	(72, 1)	-104.105
(73, 1)	-100.633	(74, 1)	-102.869	(75, 1)	-106.242	(76, 1)	-103.441
(77, 1)	-108.938	(78, 1)	-116.231	(79, 1)	-122.256	(80, 1)	-127.875

TABLE VII. (CONT)

(81, 1)	-110.712	(82, 1)	-105.763	(83, 1)	-97.2223	(84, 1)	-129.596
(85, 1)	-164.458	(86, 1)	-109.969	(87, 1)	-104.267	(88, 1)	-99.7456
(89, 1)	-108.821	(90, 1)	-113.774	(91, 1)	-117.863	(92, 1)	-111.531
(93, 1)	-104.336	(94, 1)	-95.8157	(95, 1)	-89.5109	(96, 1)	-84.4605
(97, 1)	-87.3417	(98, 1)	-92.6346	(99, 1)	-118.208	(100, 1)	-119.728
(101, 1)	-122.472	(102, 1)	-118.839	(103, 1)	-100.555	(104, 1)	-131.439
(105, 1)	-130.295	(106, 1)	-129.338	(107, 1)	-124.707	(108, 1)	-121.975
(109, 1)	-149.357	(110, 1)	-139.680	(111, 1)	-124.642	(112, 1)	-125.813
(113, 1)	-124.459	(114, 1)	-117.579	(115, 1)	-102.252	(116, 1)	-89.5380
(117, 1)	-89.5175	(118, 1)	-87.2348	(119, 1)	-90.7523	(120, 1)	-94.6675
(121, 1)	-101.449	(122, 1)	-101.951	(123, 1)	-101.253	(124, 1)	-105.456
(125, 1)	-107.060	(126, 1)	-108.054	(127, 1)	-115.905	(128, 1)	-117.568
(129, 1)	-93.6190	(130, 1)	-88.7505	(131, 1)	-84.7716	(132, 1)	-95.7142
(133, 1)	-105.480	(134, 1)	-114.317	(135, 1)	-107.008	(136, 1)	-105.219
(137, 1)	-85.8535	(138, 1)	-78.2812	(139, 1)	-73.5235	(140, 1)	-77.2531
(141, 1)	-80.8575	(142, 1)	-110.228	(143, 1)	-86.0089	(144, 1)	-63.7351
(145, 1)	-66.2388	(146, 1)	-69.7646	(147, 1)	-61.3255	(148, 1)	-62.1837
(149, 1)	-63.9774	(150, 1)	-73.9491	(151, 1)	-85.0730	(152, 1)	-96.7251
(153, 1)	-89.3836	(154, 1)	-80.9349	(155, 1)	-107.385	(156, 1)	-118.080
(157, 1)	-129.069	(158, 1)	-115.062	(159, 1)	-99.6724	(160, 1)	-145.219
(161, 1)	-164.614	(162, 1)	-178.918	(163, 1)	-157.767	(164, 1)	-135.048
(165, 1)	-139.903	(166, 1)	-150.824	(167, 1)	-161.942	(168, 1)	-164.848
(169, 1)	-168.509	(170, 1)	-171.957	(171, 1)	-174.327	(172, 1)	-177.106
(173, 1)	-99.4853	(174, 1)	-102.641	(175, 1)	-106.031	(176, 1)	-125.122
(177, 1)	-143.810	(178, 1)	-62.6696	(179, 1)	-61.0697	(180, 1)	-59.4011
(181, 1)	-73.3792	(182, 1)	-89.3711	(183, 1)	-54.9996	(184, 1)	-50.2495
(185, 1)	-45.7135	(186, 1)	-65.6818	(187, 1)	-90.0922	(188, 1)	-113.927
(189, 1)	-111.815	(190, 1)	-109.083	(191, 1)	-142.620	(192, 1)	-170.957
(193, 1)	-198.437	(194, 1)	-186.899	(195, 1)	-174.387	(196, 1)	-197.417
(197, 1)	-196.179	(198, 1)	-198.319	(199, 1)	-189.755	(200, 1)	-179.986
(201, 1)	-201.506	(202, 1)	-203.606	(203, 1)	-205.040	(204, 1)	-191.717
(205, 1)	-178.076	(206, 1)	-167.752	(207, 1)	-168.917	(208, 1)	-170.397
(209, 1)	-191.372	(210, 1)	-177.299	(211, 1)	-162.957	(212, 1)	-163.859
(213, 1)	-167.061	(214, 1)	-170.652	(215, 1)	-170.068	(216, 1)	-168.827
(217, 1)	-161.711	(218, 1)	-161.554	(219, 1)	-163.103	(220, 1)	-152.706
(221, 1)	-139.719	(222, 1)	-125.438	(223, 1)	-141.437	(224, 1)	-156.514
(225, 1)	-197.754	(226, 1)	-190.648	(227, 1)	-182.320	(228, 1)	-176.598
(229, 1)	-170.903	(230, 1)	-165.232	(231, 1)	-167.637	(232, 1)	-166.303
(233, 1)	-198.995	(234, 1)	-197.472	(235, 1)	-194.531	(236, 1)	-190.206
(237, 1)	-186.145	(238, 1)	-185.740	(239, 1)	-173.345	(240, 1)	-160.777
(241, 1)	-170.863	(242, 1)	-182.188	(243, 1)	-110.847	(244, 1)	-106.561
(245, 1)	-101.513	(246, 1)	-120.920	(247, 1)	-140.599	(248, 1)	-49.6420
(249, 1)	-52.6350	(250, 1)	-53.0969	(251, 1)	-68.6765	(252, 1)	-84.2868
(253, 1)	-60.3605	(254, 1)	-63.7238	(255, 1)	-66.8858	(256, 1)	-69.1371
(257, 1)	-72.7123	(258, 1)	-79.3820	(259, 1)	-89.4537	(260, 1)	-96.8346
(261, 1)	-102.321	(262, 1)	-126.300	(263, 1)	-151.122	(264, 1)	-154.210
(265, 1)	-157.446	(266, 1)	-151.684	(267, 1)	-157.295	(268, 1)	-162.495
(269, 1)	-167.109	(270, 1)	-175.962	(271, 1)	-185.229	(272, 1)	-164.587
(273, 1)	-162.380	(274, 1)	-160.466	(275, 1)	-167.372	(276, 1)	-175.C21
(277, 1)	-156.022	(278, 1)	-151.743	(279, 1)	-147.623	(280, 1)	-152.380
(281, 1)	-158.880	(282, 1)	-146.696	(283, 1)	-143.736	(284, 1)	-142.546
(285, 1)	-141.506	(286, 1)	-140.308	(287, 1)	-138.881	(288, 1)	-141.631
(289, 1)	-144.963	(290, 1)	-77.5630	(291, 1)	-80.2560	(292, 1)	-85.6065
(293, 1)	-104.401	(294, 1)	-123.421	(295, 1)	-79.1892	(296, 1)	-95.0839
(297, 1)	-112.717	(298, 1)	-104.491	(299, 1)	-95.8465	(300, 1)	-115.219
(301, 1)	-119.341	(302, 1)	-126.352	(303, 1)	-116.159	(304, 1)	-107.805
(305, 1)	-95.4845	(306, 1)	-91.4043	(307, 1)	-88.1141	(308, 1)	-111.455
(309, 1)	-127.560	(310, 1)	-143.789	(311, 1)	-141.594	(312, 1)	-141.190
(313, 1)	-143.327	(314, 1)	-142.947	(315, 1)	-142.638	(316, 1)	-141.827
(317, 1)	-139.524	(318, 1)	-136.801	(319, 1)	-139.346	(320, 1)	-142.517
(321, 1)	-146.264	(322, 1)	-146.540	(323, 1)	-145.247	(324, 1)	-98.3540
(325, 1)	-103.108	(326, 1)	-118.502	(327, 1)	-156.158	(328, 1)	-191.048
(329, 1)	-219.241	(330, 1)	-189.420	(331, 1)	-154.483	(332, 1)	-180.224
(333, 1)	-114.002	(334, 1)	-125.292	(335, 1)	-139.240	(336, 1)	-137.784
(337, 1)	-150.365	(338, 1)	-149.159				

TABLE VII. (CONT)

DATA SET RUN 96 ID NUMBER 159 SIZE (338, 1) H.T. ENDWALL NODE STANTON NUMBER

ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE
(1, 1)	.425424E-02	(2, 1)	.443519E-02	(3, 1)	.462453E-02	(4, 1)	.481996E-02
(5, 1)	.514308E-02	(6, 1)	.577699E-02	(7, 1)	.635177E-02	(8, 1)	.624394E-02
(9, 1)	.614766E-02	(10, 1)	.601602E-02	(11, 1)	.543720E-02	(12, 1)	.481473E-02
(13, 1)	.507321E-02	(14, 1)	.520256E-02	(15, 1)	.510357E-02	(16, 1)	.467569E-02
(17, 1)	.435960E-02	(18, 1)	.419866E-02	(19, 1)	.487236E-02	(20, 1)	.564861E-02
(21, 1)	.468374E-02	(22, 1)	.414728E-02	(23, 1)	.381606E-02	(24, 1)	.367111E-02
(25, 1)	.354682E-02	(26, 1)	.343867E-02	(27, 1)	.364384E-02	(28, 1)	.394847E-02
(29, 1)	.378335E-02	(30, 1)	.393032E-02	(31, 1)	.415762E-02	(32, 1)	.400632E-02
(33, 1)	.386569E-02	(34, 1)	.374235E-02	(35, 1)	.355762E-02	(36, 1)	.342958E-02
(37, 1)	.422358E-02	(38, 1)	.426134E-02	(39, 1)	.429692E-02	(40, 1)	.422255E-02
(41, 1)	.415318E-02	(42, 1)	.409075E-02	(43, 1)	.397926E-02	(44, 1)	.386273E-02
(45, 1)	.434707E-02	(46, 1)	.439872E-02	(47, 1)	.445432E-02	(48, 1)	.439805E-02
(49, 1)	.434434E-02	(50, 1)	.429268E-02	(51, 1)	.422691E-02	(52, 1)	.416002E-02
(53, 1)	.458836E-02	(54, 1)	.477101E-02	(55, 1)	.493919E-02	(56, 1)	.482381E-02
(57, 1)	.471891E-02	(58, 1)	.462494E-02	(59, 1)	.449148E-02	(60, 1)	.438635E-02
(61, 1)	.444818E-02	(62, 1)	.426146E-02	(63, 1)	.407350E-02	(64, 1)	.407152E-02
(65, 1)	.407431E-02	(66, 1)	.388719E-02	(67, 1)	.372028E-02	(68, 1)	.355523E-02
(69, 1)	.358786E-02	(70, 1)	.365246E-02	(71, 1)	.335424E-02	(72, 1)	.313734E-02
(73, 1)	.302681E-02	(74, 1)	.310628E-02	(75, 1)	.322202E-02	(76, 1)	.311122E-02
(77, 1)	.327897E-02	(78, 1)	.350263E-02	(79, 1)	.370656E-02	(80, 1)	.389930E-02
(81, 1)	.333367E-02	(82, 1)	.318092E-02	(83, 1)	.292359E-02	(84, 1)	.392619E-02
(85, 1)	.505891E-02	(86, 1)	.329988E-02	(87, 1)	.311743E-02	(88, 1)	.297510E-02
(89, 1)	.324701E-02	(90, 1)	.339487E-02	(91, 1)	.351803E-02	(92, 1)	.334102E-02
(93, 1)	.313192E-02	(94, 1)	.287256E-02	(95, 1)	.267603E-02	(96, 1)	.252011E-02
(97, 1)	.260482E-02	(98, 1)	.276239E-02	(99, 1)	.359495E-02	(100, 1)	.363396E-02
(101, 1)	.370943E-02	(102, 1)	.357620E-02	(103, 1)	.300939E-02	(104, 1)	.403570E-02
(105, 1)	.399926E-02	(106, 1)	.396834E-02	(107, 1)	.381358E-02	(108, 1)	.371354E-02
(109, 1)	.460636E-02	(110, 1)	.429907E-02	(111, 1)	.375786E-02	(112, 1)	.377952E-02
(113, 1)	.372744E-02	(114, 1)	.351544E-02	(115, 1)	.305155E-02	(116, 1)	.267014E-02
(117, 1)	.266860E-02	(118, 1)	.260110E-02	(119, 1)	.270645E-02	(120, 1)	.282415E-02
(121, 1)	.302809E-02	(122, 1)	.303903E-02	(123, 1)	.301737E-02	(124, 1)	.314964E-02
(125, 1)	.319601E-02	(126, 1)	.322536E-02	(127, 1)	.345506E-02	(128, 1)	.350438E-02
(129, 1)	.280449E-02	(130, 1)	.266886E-02	(131, 1)	.256442E-02	(132, 1)	.290509E-02
(133, 1)	.320772E-02	(134, 1)	.348067E-02	(135, 1)	.322030E-02	(136, 1)	.315053E-02
(137, 1)	.256490E-02	(138, 1)	.234565E-02	(139, 1)	.221627E-02	(140, 1)	.233150E-02
(141, 1)	.244300E-02	(142, 1)	.329305E-02	(143, 1)	.257120E-02	(144, 1)	.191157E-02
(145, 1)	.198954E-02	(146, 1)	.209943E-02	(147, 1)	.185319E-02	(148, 1)	.189614E-02
(149, 1)	.196835E-02	(150, 1)	.227748E-02	(151, 1)	.262639E-02	(152, 1)	.299621E-02
(153, 1)	.274220E-02	(154, 1)	.245896E-02	(155, 1)	.333670E-02	(156, 1)	.368027E-02
(157, 1)	.403504E-02	(158, 1)	.355458E-02	(159, 1)	.304439E-02	(160, 1)	.455589E-02
(161, 1)	.518472E-02	(162, 1)	.564303E-02	(163, 1)	.490905E-02	(164, 1)	.415096E-02
(165, 1)	.440298E-02	(166, 1)	.477970E-02	(167, 1)	.516870E-02	(168, 1)	.527147E-02
(169, 1)	.539137E-02	(170, 1)	.551457E-02	(171, 1)	.557004E-02	(172, 1)	.562972E-02
(173, 1)	.309736E-02	(174, 1)	.321294E-02	(175, 1)	.333749E-02	(176, 1)	.395841E-02
(177, 1)	.457124E-02	(178, 1)	.193806E-02	(179, 1)	.189725E-02	(180, 1)	.185168E-02
(181, 1)	.229104E-02	(182, 1)	.279843E-02	(183, 1)	.172232E-02	(184, 1)	.157938E-02
(185, 1)	.144385E-02	(186, 1)	.207811E-02	(187, 1)	.286098E-02	(188, 1)	.364271E-02
(189, 1)	.355835E-02	(190, 1)	.345321E-02	(191, 1)	.459094E-02	(192, 1)	.553963E-02
(193, 1)	.646717E-02	(194, 1)	.605386E-02	(195, 1)	.560768E-02	(196, 1)	.645041E-02
(197, 1)	.641442E-02	(198, 1)	.648828E-02	(199, 1)	.616758E-02	(200, 1)	.580945E-02
(201, 1)	.659971E-02	(202, 1)	.666301E-02	(203, 1)	.669504E-02	(204, 1)	.623180E-02
(205, 1)	.575561E-02	(206, 1)	.539604E-02	(207, 1)	.542900E-02	(208, 1)	.547241E-02
(209, 1)	.621802E-02	(210, 1)	.572341E-02	(211, 1)	.521979E-02	(212, 1)	.527124E-02
(213, 1)	.537837E-02	(214, 1)	.549055E-02	(215, 1)	.547415E-02	(216, 1)	.543419E-02
(217, 1)	.515098E-02	(218, 1)	.511944E-02	(219, 1)	.514584E-02	(220, 1)	.485764E-02
(221, 1)	.445813E-02	(222, 1)	.401491E-02	(223, 1)	.453686E-02	(224, 1)	.502980E-02
(225, 1)	.649451E-02	(226, 1)	.629290E-02	(227, 1)	.604593E-02	(228, 1)	.584827E-02
(229, 1)	.565135E-02	(230, 1)	.545513E-02	(231, 1)	.548617E-02	(232, 1)	.538768E-02
(233, 1)	.656023E-02	(234, 1)	.654785E-02	(235, 1)	.648568E-02	(236, 1)	.633468E-02
(237, 1)	.618820E-02	(238, 1)	.608654E-02	(239, 1)	.571332E-02	(240, 1)	.533333E-02
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

TABLE VII. (CONT)

(241, 1)	.288701E-02	(242, 1)	.807404E-02	(243, 1)	[illegible]	(244, 1)	[illegible]
(245, 1)	.332045E-02	(246, 1)	.397756E-02	(247, 1)	.464669E-02	(248, 1)	.158161E-02
(249, 1)	.168975E-02	(250, 1)	.171651E-02	(251, 1)	.222427E-02	(252, 1)	.273872E-02
(253, 1)	.197261E-02	(254, 1)	.210428E-02	(255, 1)	.223126E-02	(256, 1)	.230700E-02
(257, 1)	.242921E-02	(258, 1)	.265606E-02	(259, 1)	.296782E-02	(260, 1)	.318826E-02
(261, 1)	.344371E-02	(262, 1)	.427081E-02	(263, 1)	.512788E-02	(264, 1)	.518801E-02
(265, 1)	.525787E-02	(266, 1)	.556415E-02	(267, 1)	.532556E-02	(268, 1)	.548794E-02
(269, 1)	.562891E-02	(270, 1)	.590613E-02	(271, 1)	.619671E-02	(272, 1)	.554495E-02
(273, 1)	.546613E-02	(274, 1)	.539231E-02	(275, 1)	.560025E-02	(276, 1)	.583190E-02
(277, 1)	.524895E-02	(278, 1)	.511105E-02	(279, 1)	.497890E-02	(280, 1)	.509978E-02
(281, 1)	.528426E-02	(282, 1)	.501443E-02	(283, 1)	.495125E-02	(284, 1)	.495073E-02
(285, 1)	.491770E-02	(286, 1)	.486882E-02	(287, 1)	.480231E-02	(288, 1)	.485308E-02
(289, 1)	.492603E-02	(290, 1)	.261939E-02	(291, 1)	.273607E-02	(292, 1)	.294683E-02
(293, 1)	.360833E-02	(294, 1)	.427845E-02	(295, 1)	.267060E-02	(296, 1)	.324240E-02
(297, 1)	.388293E-02	(298, 1)	.360346E-02	(299, 1)	.330788E-02	(300, 1)	.400650E-02
(301, 1)	.418814E-02	(302, 1)	.447398E-02	(303, 1)	.412444E-02	(304, 1)	.383171E-02
(305, 1)	.338383E-02	(306, 1)	.320708E-02	(307, 1)	.306189E-02	(308, 1)	.396760E-02
(309, 1)	.455812E-02	(310, 1)	.515369E-02	(311, 1)	.501550E-02	(312, 1)	.494938E-02
(313, 1)	.514701E-02	(314, 1)	.514088E-02	(315, 1)	.513488E-02	(316, 1)	.511036E-02
(317, 1)	.505261E-02	(318, 1)	.497699E-02	(319, 1)	.510650E-02	(320, 1)	.526152E-02
(321, 1)	.544059E-02	(322, 1)	.539877E-02	(323, 1)	.529261E-02	(324, 1)	.349605E-02
(325, 1)	.367947E-02	(326, 1)	.427872E-02	(327, 1)	.558559E-02	(328, 1)	.693018E-02
(329, 1)	.803468E-02	(330, 1)	.693107E-02	(331, 1)	.562007E-02	(332, 1)	.660734E-02
(333, 1)	.410007E-02	(334, 1)	.453348E-02	(335, 1)	.493645E-02	(336, 1)	.481965E-02
(337, 1)	.511419E-02	(338, 1)	.506208E-02				

***RUN 122 ID 3 ***

DATA SET RUN 122 ID NUMBER 3 SIZE (1, 1) IDEAL EXIT MACH NUMBER

ELEMENT	VALUE
(1, 1)	.707661

***RUN 122 ID 5 ***

DATA SET RUN 122 ID NUMBER 5 SIZE (1, 1) AVERAGE INLET STATIC PRESSURE

ELEMENT	VALUE
(1, 1)	58.4169

***GEOMETRY ID 5 ***

DATA SET GEOMETRY ID NUMBER 5 SIZE (2, 42) ENDWALL STATIC PRESSURE TAP COORDINATES

ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE	ELEMENT	VALUE
(1, 1)	.750000E-01	(2, 1)	.0				
(1, 2)	.665000	(2, 2)	.0				
(1, 3)	1.33000	(2, 3)	.0				
(1, 4)	1.99500	(2, 4)	.0				
(1, 5)	2.66000	(2, 5)	.0				
(1, 6)	.274000	(2, 6)	.409000				
(1, 7)	.801000	(2, 7)	.409000				
(1, 8)	1.32800	(2, 8)	.409000				
(1, 9)	1.85500	(2, 9)	.409000				
(1, 10)	2.38200	(2, 10)	.409000				
(1, 11)	.688000	(2, 11)	.908000				
(1, 12)	1.09100	(2, 12)	.908000				
(1, 13)	1.49300	(2, 13)	.908000				
(1, 14)	1.89600	(2, 14)	.908000				
(1, 15)	2.29800	(2, 15)	.908000				
(1, 16)	1.36200	(2, 16)	1.40800				
(1, 17)	1.67600	(2, 17)	1.40800				
(1, 18)	1.99000	(2, 18)	1.40800				
(1, 19)	2.30400	(2, 19)	1.40800				
(1, 20)	2.61800	(2, 20)	1.40800				
(1, 21)	1.95000	(2, 21)	1.75800				

TABLE VII. (CONT)

(1,	22)	2.25700	(2,	22)	1.75800
(1,	23)	2.58400	(2,	23)	1.75800
(1,	24)	2.90100	(2,	24)	1.75800
(1,	25)	3.21800	(2,	25)	1.75800
(1,	26)	2.73000	(2,	26)	2.15800
(1,	27)	3.09000	(2,	27)	2.15800
(1,	28)	3.45000	(2,	28)	2.15800
(1,	29)	3.61000	(2,	29)	2.15800
(1,	30)	4.17000	(2,	30)	2.15800
(1,	31)	3.45800	(2,	31)	2.47400
(1,	32)	3.86600	(2,	32)	2.47400
(1,	33)	4.27300	(2,	33)	2.47400
(1,	34)	4.68100	(2,	34)	2.47400
(1,	35)	5.08600	(2,	35)	2.47400
(1,	36)	2.94200	(2,	36)	2.97400
(1,	37)	3.63000	(2,	37)	2.97400
(1,	36)	4.31800	(2,	38)	2.97400
(1,	39)	4.85300	(2,	39)	2.97400
(1,	40)	5.36800	(2,	40)	2.97400
(1,	41)	5.92300	(2,	41)	2.97400
(1,	42)	6.45800	(2,	42)	2.97400

***END

Function 1--Endwall Interpolation (ANTRP)

ANTRP is used to interpolate data associated with fixed endwall instrumentation. The formal input parameters for ANTRP are XIN, YIN, NRUN, IDX, and IDAT, as defined in Table VIII.

TABLE VIII. ANTRP INPUT PARAMETERS

ANTRP Parameter	Definition
XIN	Axial coordinate
YIN	Tangential coordinate
NRUN	Run number (data set)
IDX	Geometry I.D. of appropriate instrumentation coordinates
IDAT	Data array I.D. corresponding to the appropriate instrumentation

The function software will accept IDX values of 5 or less, corresponding to endwall instrumentation coordinates. ANTRP accepts IDAT values of 32, 74, 75, or 78 only, which correspond to endwall pressures, hot surface temperatures, cold surface temperatures, and adiabatic wall temperatures, respectively. ANTRP also checks the coordinates (XIN, YIN) to verify that they are within the valid endwall domain. The function returns a single interpolated value of the variable in C when the statement C = ANTRP (XIN,YIN,NRUN,IDX,IDAT) is encountered.

Function 2--Exit Plane Interpolation (BNTRP)

The function BNTRP is used to interpolate raw and reduced data associated with the traversing exit probe. The function inputs are ZIN, YIN, NRUN, IDAT, as defined in Table IX.

TABLE IX. BNTRP INPUT PARAMETERS

BNTRP Parameter	Definition
ZIN	Radial coordinate
YIN	Tangential coordinate
NRUN	Run number (data set)
IDAT	Exit data array I.D. (array dimensioned (25, 9))

The function software accepts IDAT values of 90 through 107 only, corresponding to exit data arrays. BNTRP also checks the coordinates (ZIN, YIN) to verify that they are within the exit plane traversed for the run NRUN. The probe position coordinates are retrieved automatically by BNTRP. BNTRP returns a single interpolated value of the variable in C when the statement C = BNTRP (ZIN,YIN,NRUN,IDAT) is encountered.

The listing of a program that demonstrates the use of ANTRP and BNTRP is given in Table X. The program is loaded with the data base access subroutines and interpolation subroutines, and reads cards specifying an interpolation function and input parameters until an END card is encountered. The program reads input parameters from the cards, synthesizes the correct call statement of ANTRP or BNTRP, and prints the resulting interpolated value for each data card. A sample input data set is shown in Table XI and the resulting output appears in Table XII. This program also uses the access subroutines DBDMNS, DBRUN, DBSET, and DBFCHC to retrieve the ANTRP input data and coordinate arrays, and displays them for convenience.

3.1.3 File Access Subroutines

The access subroutines are the building blocks of FORTRAN programs interacting with the data base. They read and write sequential magnetic tape files and create and access a random-access data bank disk file. Several routines interact with the random-access data bank, performing the specialized tasks required for item storage and retrieval. Table XIII summarizes tasks performed by the access subroutines. A description and usage instructions are given below for each subroutine, followed by comments on the general structure for data base programs in Section 3.1.4.

Subroutine DBLOAD

The calling convention for using DBLOAD is simply "CALL DBLOAD". DBLOAD is called to read the data base from a sequential tape file and generate a random-access data bank disk file. The subroutine begins with a call to DBSTRT, which initializes the master index for the random-access file on unit 29 and writes the first record of the file. Records are written to and read from the random-access data bank using the general routines DBPUT and DBGET, respectively. The subroutines DBSTRT, DBPUT, and DBGET are user-transparent. After the data bank is initialized, records are read from the sequential tape file using DBREAD, another user-transparent routine. The records are then stored in the random-access data bank, where a global directory and set directories map the location of each datum for future reference. Upon completion of the loading, DBDONE is called within DBLOAD to close the data bank and write its master index.

The subroutine DBLOAD expects to find a sequential tape file of the data base on unit 27. TAPE27 and TAPE29 must be specified in the program statement. The output resulting from a call to DBLOAD is shown in Table XIV. The number of data sets and the number of variables stored for each data set are listed. Upon successful completion of loading, the final message is printed.

TABLE X. INTERPOLATION SOFTWARE DEMONSTRATION PROGRAM LISTING

```
C     MAINLINE TO DEMONSTRATE USE OF FUNCTIONS ANTRP AND BNTRP
C
      LOGICAL SCAN
      DIMENSION CARD(20),DATA(5),XCOORD(300),XDATA(150)
      DATA NF/5/
C
      CALL DELOAD
      CALL DEOPEN
C
    1 READ(NR,2,END=99) CARD
    2 FORMAT(20A4)
      IF(SCAN(CARD,'ANTRP$')) GO TO 20
      IF(SCAN(CARD,'BNTRP$')) GO TO 30
      IF(SCAN(CARD,'END$')) GO TO 99
      IF(SCAN(CARD,' $')) GO TO 1
      WRITE(6,10)
   10 FORMAT(' COMMAND ERROR')
      GO TO 1
C
C     ANTRP DATA
   20 L=1
      DO 25 I=1,5
      DATA(I)=FGTNUM(CARD,L,80,L,NF)
      IF(NF.EQ.0) GO TO 97
   25 CONTINUE
      XIN=DATA(1)
      YIN=DATA(2)
      IDX=DATA(3)
      NRUN=DATA(4)
      IDATA=DATA(5)
      CALL DERUN(NRUN,2)
      CALL DEDMNS(IDATA,MP,MS)
      CALL DEFCHC(IDATA,MP,MS,XDATA)
      CALL DESET('GEOMETRY',2)
      CALL DEFCHC(IDX,2,MP,XCOORD)
      WRITE(6,21)
   21 FORMAT(1H0,16HANTRP INPUT DATA,//,5X1HX,5X1HY,8X4HDATA)
      DO 22 I=1,MP
      K=2*I-1
      KP1=K+1
   22 WRITE(6,24) XCOORD(K),XCOORD(KP1),XDATA(I)
   24 FORMAT(2F7.2,F12.2)
      A=ANTRP(XIN,YIN,NRUN,IDX,IDATA)
      WRITE(6,26) XIN,YIN,NRUN,IDX,IDATA,A
   26 FORMAT(/,' ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA=',2F7.2,3I5,/,
     1  ' OUTPUT VALUE=',F10.3)
      GO TO 1
C
C      BNTRP DATA
   30 L=1
      DO 35 I=1,4
      DATA(I)=FGTNUM(CARD,L,80,L,NF)
      IF(NF.EQ.0) GO TO 97
   35 CONTINUE
      ZIN=DATA(1)
      YIN=DATA(2)
      NRUN=DATA(3)
      IDATA=DATA(4)
      B=BNTRP(ZIN,YIN,NRUN,IDATA)
      WRITE(6,36) ZIN,YIN,NRUN,IDATA,B
   36 FORMAT(' BNTRP INPUT ZIN,YIN,NRUN,IDATA=',2F7.2,2I5,/,
     1  ' OUTPUT VALUE=',F10.3)
      GO TO 1
   97 WRITE(6,98)
   98 FORMAT(' MISSING INPUT PARAMETER')
      GO TO 1
   99 CALL DEDONE
      CALL EXIT
      END
```

TABLE XI. SAMPLE INPUT FOR INTERPOLATION DEMONSTRATION PROGRAM

```
ANTRP  2.332 .5717 1 122 74
ANTRP  2.332 .5717 4 122 78
ANTRP  2.332 .5717 5 122 32
ANTRP  5.5 2.5933 1 122 74
ANTRP  5.5 2.5933 4 122 78
ANTRP  5.5 2.5933 5 122 32
BNTRP  1.200 12.0  122 90
END
```

Subroutine DBOPEN

DBOPEN opens an existing data bank on unit 29 by reading its first global directory record and reading its master index into core memory. Once the data bank has been opened using DBOPEN, data may be accessed using the subroutines DBSET, DBRUN, DBSTDM, DBDMNS, DBFCHC, DBFCHP, DBSTRC, and DBSTRP. The calling convention is simply "CALL DBOPEN." No printed output is generated by this routine.

Subroutine DBSET (NAME,IOPT)

Subroutine DBSET is used to access, create, or delete a data set in the random-access data bank. The subroutine calling convention is "CALL DBSET (NAME, IOPT)." NAME is the eight-character data set identifier, such as GEOMETRY or RUNXXXXX. IOPT is an integer corresponding to one of the desired action options outlined in Table XV.

Option 1 is useful for creating new data sets. Option 2 is used for accessing data sets only. Option 3 is used to ensure a unique new data set is being formed. For example, option 3 is used when the data bank is loaded using DBLOAD. Option 4 is used to purge data sets no longer needed.

Subroutine DBSET generates no printed output when it successfully performs the user's desired task. If option 2 is elected and the data set is not found, the message "DATA SET NAME NOT FOUND" is printed. If option 3 is elected and the data set is found, the message "DATA SET NAME ALREADY EXISTS" is printed. If the message "DATA BANK IS INCONSISTENT" is printed, an error occurred when the data bank was loaded. This message indicates a software problem in DBLOAD and should not appear during normal operation of the data base.

Subroutine DBRUN (NRUN,ICODE)

Subroutine DBRUN synthesizes the data set name RUNXXXXX and calls DBSET to access the data set. The calling convention is "CALL DBRUN (NRUN,ICODE)". NRUN is the experimental run number associated with a particular data set. ICODE is the user option associated with the DBSET call as outlined in Table XV.

TABLE XII. SAMPLE OUTPUT FOR INTERPOLATION DEMONSTRATION PROGRAM

CANTRP INPUT DATA

X	Y	DATA
0.0	0.0	549.90
0.29	0.0	543.10
1.77	0.0	-1.00
2.27	0.0	538.60
2.80	0.0	538.70
2.10	0.23	-1.00
0.23	0.41	569.50
0.38	0.41	552.00
1.02	0.41	549.30
2.43	0.41	551.00
1.92	0.55	552.90
1.10	0.78	566.40
1.50	0.77	560.10
2.30	0.78	570.30
0.31	1.00	567.50
1.04	1.00	571.30
1.30	1.00	572.00
2.39	1.00	-1.00
1.10	1.35	588.50
1.70	1.38	588.20
2.20	1.41	586.90
2.73	1.41	591.30
1.74	1.66	605.70
2.17	1.66	602.00
2.62	1.60	-1.00
3.00	1.66	595.70
2.09	1.66	-1.00
2.43	2.00	-1.00
3.00	1.62	605.70
3.55	1.88	596.20
2.55	1.87	601.50
3.04	2.00	-1.00
3.49	2.00	607.40
4.01	2.00	598.50
2.90	2.28	612.30
3.42	2.25	-1.00
3.99	2.25	608.70
4.40	2.25	605.80
3.01	2.47	603.50
3.34	2.47	607.20
4.58	2.47	613.90
3.96	2.47	611.30
5.20	2.47	623.00
3.30	2.56	614.80
3.79	2.71	611.00
4.48	2.71	614.00
5.16	2.71	-1.00
2.94	2.97	-1.00
3.82	2.97	601.90
4.30	2.97	606.10
5.07	2.97	-1.00
5.85	2.97	642.60
6.63	2.97	657.80

ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA= 2.33 0.57 122 1 74
OUTPUT VALUE= 557.549

TABLE XII. (CONT)

OANTFP INPUT DATA

X	Y	DATA
5.82	3.01	722.70
5.16	3.01	716.00
4.75	3.01	714.90
4.29	3.01	717.40
3.84	3.01	717.20
3.40	3.01	716.10
2.95	3.01	-1.00
5.82	2.75	724.70
5.16	2.75	723.20
4.75	2.75	-1.00
4.29	2.75	722.70
3.84	2.75	723.30
3.40	2.75	716.90
2.95	2.75	720.00
5.37	2.49	-1.00
[illegible]	2.49	-1.00
[illegible]	2.49	[illegible]
[illegible]	2.49	-1.00
4.34	2.49	[illegible]
[illegible]	2.49	724.00
3.37	2.49	719.20
3.09	2.49	720.70
4.90	2.36	737.50
4.74	2.36	734.50
[illegible]	2.36	-1.00
[illegible]	2.36	[illegible]
[illegible]	2.23	-1.00
4.40	2.23	747.50
4.15	2.23	733.90
3.90	2.23	720.10
3.50	2.23	727.10
3.10	2.23	716.70
[illegible]	2.23	-1.00
4.25	2.10	725.90
4.05	2.10	-1.00
3.82	2.10	722.10
[illegible]	1.97	-1.00
3.72	1.97	723.70
3.30	1.97	721.90
3.06	1.97	[illegible]
2.62	1.97	718.00
2.21	1.97	-1.00
3.60	1.84	739.00
3.32	1.71	-1.00
2.95	1.71	722.30
2.55	1.71	716.50
2.15	1.71	-1.00
1.79	1.71	-1.00
2.85	1.45	-1.00
2.49	1.45	716.90
2.10	1.45	716.60
1.72	1.45	726.00
1.36	1.45	-1.00
2.53	1.19	-1.00
2.16	1.19	717.70
1.76	1.19	715.60
1.36	1.19	716.90

TABLE XII. (CONT)

0.98	1.19	-1.00
2.31	1.06	-1.00
2.41	0.93	709.00
2.09	0.93	718.20
1.73	0.93	-1.00
1.37	0.93	709.10
1.02	0.93	-1.00
0.69	0.93	713.00
2.24	0.80	725.40
2.40	0.67	-1.00
2.05	0.67	720.30
1.63	0.67	711.10
1.24	0.67	-1.00
0.84	0.67	-1.00
0.47	0.67	715.50
2.35	0.54	-1.00
2.32	0.41	-1.00
2.22	0.41	720.60
1.39	0.41	-1.00
1.07	0.41	707.50
[illegible]	0.41	-1.00
0.91	0.41	710.20
0.38	0.41	709.00
0.25	0.41	702.40
2.42	0.28	-1.00
2.00	0.28	724.10
2.60	0.15	737.90
2.30	0.15	734.00
1.95	0.15	720.20
1.59	0.15	706.00
1.00	0.15	705.50
0.32	0.15	701.70
-0.01	0.15	-1.00

ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA= 2.35 0.57 122 4 76
OUTPUT VALUE= 723.772
CANTRP INPUT DATA

X	Y	DATA
0.07	0.0	56.65
0.57	0.0	-1.00
1.33	0.0	58.29
1.95	0.0	57.97
2.60	0.0	56.53
0.27	0.41	56.98
0.80	0.41	55.73
1.33	0.41	56.04
1.85	0.41	57.22
2.38	0.41	56.51
0.69	0.91	58.77
1.09	0.91	58.01
1.49	0.91	57.15
1.90	0.91	54.71
2.30	0.91	50.37
1.36	1.41	56.73
1.68	1.41	55.24
1.99	1.41	52.81
2.30	1.41	49.35
2.62	1.41	44.04
1.95	1.76	52.76
2.27	1.76	-1.00
2.58	1.76	45.75
2.90	1.76	[illegible]

TABLE XII. (CONT)

[illegible]	[illegible]	[illegible]
3.22	1.76	41.26
2.73	2.16	44.01
3.09	2.16	42.77
3.45	2.16	42.56
3.81	2.16	42.37
4.17	2.16	42.53
3.46	2.47	44.02
3.87	2.47	43.07
4.27	2.47	42.79
4.68	2.47	43.17
5.09	2.47	44.52
2.94	2.97	43.47
3.[illegible]	2.97	43.84
4.32	2.97	43.37
4.85	2.97	43.25
5.39	2.97	43.22
5.92	2.97	43.23
6.46	2.97	43.02

ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA= 2.55 0.57 122 5 [illegible]
OUTPUT VALUE= 5[illegible].176
CANTRP INPUT DATA

X	Y	DATA
0.0	0.0	549.90
0.89	0.0	543.10
1.77	0.0	-1.00
2.27	0.0	536.60
2.66	0.0	536.70
2.13	0.23	-1.00
0.23	0.41	569.50
0.96	0.41	553.00
1.69	0.41	549.30
2.43	0.41	551.00
1.92	0.55	552.90
1.18	0.78	566.40
1.50	0.77	560.10
2.30	0.78	570.00
0.81	1.06	567.50
1.34	1.06	573.30
1.86	1.06	572.00
2.39	1.06	-1.00
1.18	1.35	583.60
1.76	1.38	586.20
2.26	1.41	586.90
2.73	1.41	591.00
1.71	1.66	606.70
2.17	1.66	602.00
2.62	1.66	-1.00
3.08	1.66	595.70
2.09	1.86	-1.00
2.45	2.06	-1.00
3.00	1.82	605.70
3.55	1.66	596.20
2.53	1.87	601.50
3.04	2.06	-1.00
3.49	2.06	607.40
4.01	2.06	598.50
2.90	2.28	612.30
3.42	2.25	-1.00
3.99	2.25	608.70
4.48	2.25	605.80

TABLE XII. (CONT)

```
    3.01     2.47        603.50
    3.34     2.47        607.20
    4.30     2.47        613.90
    3.90     2.47        611.30
    5.20     2.47        623.00
    3.30     2.56        614.80
    3.79     2.71        611.00
    4.40     2.71        614.00
    5.13     2.71         -1.00
    2.94     2.97         -1.00
    3.62     2.97        601.90
    4.30     2.97        606.10
    5.07     2.97         -1.00
    5.35     2.97        646.60
    6.03     2.97        657.80

 WNTRP INPUT XIN,YIN,NRUN,IDX,IDATA=    3.50     2.59   122     1    74
 OUTPUT VALUE=     630.420
CANTRP INPUT DATA

     X        Y          DATA
    5.32     3.01        722.70
    5.16     3.01        718.00
    4.73     3.01        714.90
    4.29     3.01        717.40
    3.84     3.01        717.20
    3.40     3.01        718.10
    2.95     3.01         -1.00
    5.62     2.75        724.70
    5.13     2.75        723.20
    4.73     2.75         -1.00
    4.29     2.75        722.70
    3.84     2.75        723.30
    3.40     2.75        718.90
    2.95     2.75        720.60
    5.39     2.49         -1.00
    5.10     2.49         -1.00
    4.83     2.49        730.90
    4.59     2.49         -1.00
    4.34     2.49        721.60
    4.08     2.49        724.00
    3.57     2.49        719.20
    3.09     2.49        726.70
    4.96     2.36        737.30
    4.74     2.36        734.50
    4.49     2.36         -1.00
    4.24     2.36        720.40
    4.62     2.23         -1.00
    4.40     2.23        747.60
    4.15     2.23        733.90
    3.90     2.23        720.10
    3.66     2.23        727.10
    3.16     2.23        718.70
    2.69     2.23         -1.00
    4.25     2.10        725.90
    4.05     2.10         -1.00
    3.82     2.10        722.10
    3.91     1.97         -1.00
    3.72     1.97        723.70
    3.50     1.97        721.90
    3.06     1.97        718.90
    3.62     1.97        718.80
```

TABLE XII. (CONT)

```
  [illegible]  [illegible]  [illegible]
  2.21    1.97      -1.00
  3.60    1.84     739.00
  3.32    1.71      -1.00
  2.95    1.71     722.30
  2.55    1.71     716.80
  2.15    1.71      -1.00
  1.79    1.71      -1.00
  2.85    1.45      -1.00
  2.49    1.45     716.90
  2.10    1.45     716.30
  1.72    1.45     726.00
  1.36    1.45      -1.00
  2.55    1.19      -1.00
  2.16    1.19     717.70
  1.76    1.19     715.60
  1.36    1.19     716.90
  0.96    1.19      -1.00
  2.31    1.06      -1.00
  2.41    0.93     709.00
  2.39    0.93     718.20
  1.73    0.93      -1.00
  1.37    0.93     709.10
  1.02    0.93      -1.00
  0.69    0.93     713.00
  2.24    0.80     725.40
  2.40    0.67      -1.00
  2.03    0.67     720.30
  1.65    0.67     711.10
  1.24    0.67      -1.00
  0.84    0.67      -1.00
  0.47    0.67     715.50
  2.33    0.54      -1.00
  2.53    0.41      -1.00
  2.22    0.41     726.80
  1.89    0.41      -1.00
  1.57    0.41     707.50
  1.24    0.41      -1.00
  0.91    0.41     710.20
  0.58    0.41     709.00
  0.29    0.41     702.40
  2.42    0.28      -1.00
  2.06    0.28     724.10
  2.66    0.15     737.90
  2.30    0.15     734.00
  1.95    0.15     720.20
  1.59    0.15     708.00
  1.06    0.15     705.50
  0.52    0.15     701.70
 -0.01    0.15      -1.00
ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA=   5.50   2.59  122    4   76
OUTPUT VALUE=    730.498
CANTRP INPUT DATA
```

TABLE XII. (CONT)

X	Y	DATA
0.07	0.0	56.63
0.67	0.0	-1.00
1.33	0.0	56.29
1.99	0.0	57.97
2.66	0.0	58.53
0.27	0.41	58.98
0.80	0.41	56.73
1.33	0.41	56.04
1.85	0.41	57.22
2.38	0.41	56.51
0.99	0.91	56.77
1.09	0.91	58.01
1.49	0.91	57.15
1.90	0.91	54.71
2.30	0.91	50.37
1.30	1.41	50.73
1.[illegible]	1.41	55.24
1.99	1.41	52.81
2.30	1.41	49.55
2.62	1.41	44.04
1.95	1.70	52.76
2.27	1.70	-1.00
2.56	1.76	45.75
2.90	1.70	42.65
3.22	1.70	41.26
2.73	2.10	44.01
3.09	2.10	42.77
3.45	2.10	42.56
3.81	2.10	42.37
4.17	2.10	42.53
3.46	2.47	44.02
3.87	2.47	43.07
4.27	2.47	42.79
4.68	2.47	43.17
5.09	2.47	44.52
2.94	2.97	43.47
3.83	2.97	43.64
4.32	2.97	43.37
4.85	2.97	43.23
5.39	2.97	43.22
5.92	2.97	43.23
6.46	2.97	43.02

```
ANTRP INPUT XIN,YIN,NRUN,IDX,IDATA=    5.50    2.59  122    5    32
OUTPUT VALUE=     44.536
BNTRP INPUT ZIN,YIN,NRUN,IDATA=    1.20  12.00  122   90
OUTPUT VALUE=     59.037
```

TABLE XIII. ACCESS SUBROUTINE FUNCTIONS

Task	Subroutine
o Load sequential tape file into random-access data bank disk file	DBLOAD
o Open existing random-access data bank	DBOPEN
o Access/create/delete a data set	DBSET
o Access/create/delete data set "RUNXXXXX"	DBRUN
o Reserve storage for an array	DBSTDM
o Retrieve dimensions of a stored array	DBDMNS
o Store/overstore an entire array	DBSTRC
o Retrieve an entire array	DBFCHC
o Store/overstore part of an array	DBSTRP
o Retrieve part of an array	DBFCHP
o Close random-access data bank	DBDONE
o Write data bank contents to sequential tape file	DBSAVE

As with the subroutine DBSET, there is no printed output for normal execution of subroutine DBRUN. User error messages for a given option are printed as generated in DBSET.

Subroutine DBSTDM (IV,MP,MS)

DBSTDM is used to reserve storage in the random-access data bank for a matrix dimensioned MP x MS. A call to DBSTDM must be preceded by a call to DBSET or DBRUN, which accesses a particular data set.

The calling convention for this subroutine is "CALL DBSTDM (IV,MP,MS)". The user-supplied arguments of DBSTDM are variable identification number (IV), matrix primary dimension (MP), and matrix secondary dimension (MS). The subroutine adds the variable I.D. to the set directory of the data set accessed, reserves storage records for a data array dimensioned MP by MS, and initializes the data array members to zero in anticipation of partial storage of the matrix. The subroutine normally generates no printed output. If the DBSTDM call is not preceded by a DBSET or DBRUN call, the message "NO SET CALLED BEFORE DBSTDM" is printed.

Subroutine DBDMNS (IV,MP,MS)

Subroutine DBDMNS retrieves the dimensions of a matrix stored in the random-access data bank. A call to DBDMNS must be preceded by a call to DBSET or DBRUN.

The subroutine call statement is "CALL DBDMNS (IV,MP,MS)," where IV is the user-supplied variable I.D. The subroutine returns the primary and secondary dimensions associated with IV in MP and MS, respectively. Printed output is generated only if an error occurs. If the DBDMNS call is not preceded by a DBSET or DBRUN call, the message "NO SET CALLED BEFORE DBDMNS" is printed. If the variable number IV is not present in the current set directory, the message "UNABLE TO LOCATE IV IN SET NAME" is printed.

TABLE XIV. PRINTED OUTPUT FOR DBLOAD

1 45 SETS BEING LOADED IN BANK
SET RUN 122 HAS 139 VARIABLES
~~SET RUN 116 HAS 121 VARIABLES~~
SET RUN 91 HAS 113 VARIABLES
SET RUN 96 HAS 21 VARIABLES
SET RUN 93 HAS 21 VARIABLES
~~SET RUN 94 HAS 43 VARIABLES~~
SET GEOMETRY HAS 14 VARIABLES
SET RUN 95 HAS 22 VARIABLES
SET RUN 98 HAS 43 VARIABLES
~~SET RUN 99 HAS 113 VARIABLES~~
SET RUN 105 HAS 95 VARIABLES
SET RUN 107 HAS 95 VARIABLES
SET RUN 108 HAS 95 VARIABLES
~~SET RUN 109 HAS 113 VARIABLES~~
SET RUN 111 HAS 113 VARIABLES
SET RUN 112 HAS 113 VARIABLES
SET RUN 113 HAS 113 VARIABLES
~~SET RUN 57 HAS 94 VARIABLES~~
SET RUN 61 HAS 94 VARIABLES
SET RUN 67 HAS 94 VARIABLES
SET RUN 114 HAS 113 VARIABLES
~~SET RUN 123 HAS 121 VARIABLES~~
SET RUN 125 HAS 10 VARIABLES
SET RUN 131 HAS 40 VARIABLES
SET RUN 132 HAS 51 VARIABLES
~~SET RUN 133 HAS 51 VARIABLES~~
SET RUN 149 HAS 51 VARIABLES
SET RUN 150 HAS 51 VARIABLES
SET RUN 165 HAS 51 VARIABLES
~~SET RUN 166 HAS 51 VARIABLES~~
SET RUN 168 HAS 51 VARIABLES
SET RUN 169 HAS 51 VARIABLES
SET RUN 170 HAS 51 VARIABLES
~~SET RUN 171 HAS 51 VARIABLES~~
SET RUN 172 HAS 51 VARIABLES
SET RUN 173 HAS 51 VARIABLES
SET RUN 174 HAS 51 VARIABLES
~~SET RUN 118 HAS 51 VARIABLES~~
SET RUN 124 HAS 121 VARIABLES
SET RUN 86 HAS 93 VARIABLES
SET RUN 87 HAS 93 VARIABLES
~~SET RUN 89 HAS 93 VARIABLES~~
SET RUN 176 HAS 9 VARIABLES
SET RUN 177 HAS 9 VARIABLES
SET RUN 179 HAS 9 VARIABLES
~~0BANK LOAD COMPLETED~~

Subroutine DBSTRC (IV,MP,MS,ARRAY)

Subroutine DBSTRC is used to store or overstore in the data bank the entire matrix dimensioned MP by MS. The statement "CALL DBSTRC (IV,MP,MS,ARRAY)" must supply the arguments IV, MP, MS, and ARRAY. IV is the variable identification number (I.D.) to be found in the set directory or added to the set directory, and MP and MS are the primary and secondary dimensions of the data matrix, respectively. The data are stored in the vector ARRAY. ARRAY must contain MP x MS items or the storage will not be complete. The data should be arranged in ARRAY with the primary index varying fastest. In other words, elements of a matrix to be stored in the data bank under IV sized 3 by 2 should be ordered (1,1), (2,1), (3,1), (1,2), (2,2), (3,2) in ARRAY prior to the DBSTRC call.

TABLE XV. DBSET/DBRUN OPTIONS

IOPT	Action
1	Use data set if found, create if not found
2	Use data set if found, error if not found
3	Error if data set found, create if not found
4	Delete data set if found, return if not found

No printed output is generated by subroutine DBSTRC normally. If the call was not preceded by a DBSET or DBRUN call, an error message is printed.

Subroutine DBFCHC (IV,MP,MS,ARRAY)

Subroutine DBFCHC retrieves an entire matrix from the data bank. The user-supplied arguments are IV, MP, and MS, which are the variable I.D. and matrix primary and secondary dimensions, respectively. The subroutine returns the data in ARRAY, with elements ordered with the primary index varying fastest.

For a normal fetch of a complete matrix, no print is generated. If the user-supplied primary and secondary dimensions do not match the dimensions stored in the data bank for IV, the message "DIMENSIONS DON'T MATCH" is printed. If the variable I.D. (IV) is not present in the current set directory, "UNABLE TO LOCATE IV IN SET NAME" is printed. If the DBFCHC call was not preceded by a DBSET or DBRUN call, "NO SET CALLED BEFORE DBFCHC" is printed.

Subroutine DBSTRP (IV,IBP,IEP,IBS,IES,ARRAY,IPI)

Subroutine DBSTRP is used to store or overstore a part of a data matrix in the data bank. The calling sequence arguments IV, IBP, IEP, IBS, IES, ARRAY, and IPI are supplied by the user. IV is the variable identification number. IBP and IEP are the beginning and ending primary indices for the data to be stored. IBS and IES are the beginning and ending secondary indices. The data must be present in core in the vector ARRAY. The primary index IPI of the dimensioned vector ARRAY must also be supplied.

DBSTRP generates no print for a successful partial store of a vector. If the variable IV has not been initialized by a DBSTRC or DBSTDM call, the message "UNABLE TO LOCATE IV IN SET NAME" is printed. If the DBSTRP call was not preceded by a call to DBSET or DBRUN, the message "NO SET CALLED BEFORE DBSTRP" is printed.

Subroutine DBFCHP (IV,IBP,IEP,IBS,IES,ARRAY,IPI)

Subroutine DBFCHP is used to retrieve a part of a matrix stored in the data bank. The arguments IV, IBP, IEP, IBS, IES, ARRAY, and IPI are defined as for subroutine DBFCHP described above. The part of the matrix requested is returned in ARRAY, which has the primary dimension IPI.

Subroutine DBFCHP generates no print for normal execution. If the variable IV has not been previously stored, the message "UNABLE TO LOCATE IV IN SET NAME" is printed. If the DBFCHP call was not preceded by a DBSET or DBRUN call, an error message is printed.

Subroutine DBDONE

Subroutine DBDONE is used to close the random-access data bank file on unit 29. The routine writes the total number of records in the bank on the first record of the file and copies the master index onto the end of the file.

Subroutine DBSAVE

Subroutine DBSAVE is called when the random-access data bank has been edited and a new sequential tape file of the data base is to be created. The subroutine reads each record of each data set, one data set at a time, and writes the data to unit 28 using DBSTOR. DBSTOR is a user-transparent routine that packs the data into 80-character records and writes the records to unit 28 as they are filled.

A call to DBSAVE requires no input information. The subroutine assumes that a consistent data bank exists on unit 29 and that it may write to unit 28. Both TAPE28 and TAPE29 must be present in the program statement. The printed output generated by DBSAVE is similar to that of DBLOAD and is shown in Table XVI. The total number of data sets in the data bank is printed, and each set stored is listed with the number of variables in it specified.

3.1.4 User-Written Analysis/Report Programs

All FORTRAN programs interacting with the data base follow a simple format, which can be seen by referring to Table XVII or a listing of the maintenance program. First, the program statement should designate all file units used by the program. Then the data base must be loaded into a random-access disk file by calling DBLOAD. If the data bank exists as a random-access disk file designated unit 29, the DBLOAD call may be omitted. The random-access file is opened by calling DBOPEN, which retrieves the first global directory record.

Once the bank has been opened, it may be accessed by means of the subroutines described in Section 3.1.3 and the interpolation functions described in Section 3.1.2. To enable the print messages associated with the access subroutines, the print flag IERPRT must be set to 1 in the FORTRAN code and the following common block must be present in the program:
COMMON/BUFR/A(20), LREC, IA, IB, IAS, IERROR, IERPRT.

A program interacting with the data base should always be terminated with a call to DBDONE to preserve the master index for the disk file. If a new sequential file of the data base needs to be written, the DBDONE call should be followed by a DBSAVE call.

TABLE XVI. PRINTED OUTPUT FOR DBSAVE

```
 END OF UPDATES
1    45 SETS BEING STORED FROM BANK
 SET RUN    122 HAS  139 VARIABLES
 SET RUN    115 HAS  121 VARIABLES
 SET RUN     91 HAS  113 VARIABLES
 SET RUN     96 HAS   21 VARIABLES
 SET RUN     93 HAS   21 VARIABLES
 SET RUN     94 HAS   43 VARIABLES
 SET GEOMETRY   HAS   14 VARIABLES
 SET RUN     95 HAS   22 VARIABLES
 SET RUN     98 HAS   43 VARIABLES
 SET RUN     99 HAS  113 VARIABLES
 SET RUN    105 HAS   95 VARIABLES
 SET RUN    107 HAS   95 VARIABLES
 SET RUN    108 HAS   95 VARIABLES
 SET RUN    109 HAS  113 VARIABLES
 SET RUN    111 HAS  113 VARIABLES
 SET RUN    112 HAS  113 VARIABLES
 SET RUN    113 HAS  113 VARIABLES
 SET RUN     57 HAS   94 VARIABLES
 SET RUN     61 HAS   94 VARIABLES
 SET RUN     67 HAS   94 VARIABLES
 SET RUN    114 HAS  113 VARIABLES
 SET RUN    123 HAS  121 VARIABLES
 SET RUN    125 HAS   10 VARIABLES
 SET RUN    131 HAS   40 VARIABLES
 SET RUN    132 HAS   51 VARIABLES
 SET RUN    133 HAS   51 VARIABLES
 SET RUN    149 HAS   51 VARIABLES
 SET RUN    150 HAS   51 VARIABLES
 SET RUN    165 HAS   51 VARIABLES
 SET RUN    166 HAS   51 VARIABLES
 SET RUN    168 HAS   51 VARIABLES
 SET RUN    169 HAS   51 VARIABLES
 SET RUN    170 HAS   51 VARIABLES
 SET RUN    171 HAS   51 VARIABLES
 SET RUN    172 HAS   51 VARIABLES
 SET RUN    173 HAS   51 VARIABLES
 SET RUN    174 HAS   51 VARIABLES
 SET RUN    119 HAS   51 VARIABLES
 SET RUN    124 HAS  121 VARIABLES
 SET RUN     86 HAS   93 VARIABLES
 SET RUN     87 HAS   93 VARIABLES
 SET RUN     89 HAS   93 VARIABLES
 SET RUN    176 HAS    9 VARIABLES
 SET RUN    177 HAS    9 VARIABLES
 SET RUN    179 HAS    9 VARIABLES
OBANK STORAGE COMPLETED
```

3.2 GENERAL OPERATING PROCEDURES

The access subroutines, interpolation software, and utility subroutines SCAN, COMPAR, FGTNUM, and GETNUM should all be loaded into a permanent library. Sample control cards used to generate a library named BANK are shown in Table XVIII. The last subroutine in the source file should be DBDONE. If the source code is stored as a disk file, named FILE, the control cards should be preceded by an appropriate control card to access the file, and the FTN control card should be changed to read FTN,I=FILE,R=2. In such a case the source deck is omitted.

TABLE XVII. GENERAL FORMAT FOR DATA BASE PROGRAMS

```
Col. 7

      PROGRAM NAME(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE27,TAPE28,TAPE29)
      COMMON/BUFR/A(40),LREC,IA,IB,IAS,IERROR,IERPRT
      CALL DBLOAD
      CALL DBOPEN
      IERPRT = 1
     [FORTRAN code]
      CALL DBDONE
      CALL DBSAVE
      END
```

TABLE XVIII. CONTROL CARDS FOR CREATING A LIBRARY OF BANK ROUTINES

```
[JOBCARD]
FTN,R=2.
REWIND, LGO.
REQUEST,BANK,*PF.
EDITLIB.
CATALOG,BANK,RP=999.
7/8/9 [eor]
[Source deck]
7/8/9 [eor]
LIBRARY(BANK,NEW)
ADD (*+DBDONE,LGO)
FINISH.
ENDRUN.
6/7/8/9 [eof]
```

The general control card format for running MAINT, BLIST, or other data base FORTRAN programs is shown in Table XIX. Again, if the program source code is stored as a disk file called SFILE, an appropriate control card must precede the FTN card and the FTN card should be written FTN,I=SFILE,R=2. The source deck would then be omitted.

TABLE XIX. CONTROL CARDS FOR COMPILATION/EXECUTION OF DATA BASE PROGRAMS

```
[JOBCARD (INCLUDES TAPE SPECIFICATION)]

FTN,R.

ATTACH,BANK.

LIBRARY,BANK.

VSN,TAPE27=nnn,TAPE28=mmm.

REQUEST,TAPE27,PE,NORING.

REQUEST,TAPE28,PE,RING.

LGO.

7/8/9 [eor]

[Source deck]

7/8/9 [eor]

[Commands/data]

6/7/8/9 [eof]
```

The program is loaded with the data base library BANK, and serial numbers for tapes 27 and 28 are given. Unit 27 is specified NORING to guard against writing over the input data base. The source deck is followed by a sequence of commands for the programs MAINT and BLIST or input data for other programs.

To implement the data base system on an IBM 370 computer, several software changes must be made. First, the file access subroutines and utility subroutines listed in the Appendix must replace the CDC subroutines with the same names. Also, the error messages printed by the subroutines DBDMNS, DBFCHC,

DBSTRP, and DBFCHP must be changed to print the data set name correctly. The CDC versions of the routines store the entire set name as NA(1). The IBM versions store the set name in A(1) and A(2). Thus, write statements that display the set name must write A(1) and A(2), rather than NA(1). The formats should correspondingly be changed from A10 to 2A4 to display the set name.

The matrix routines used by the interpolation software contain DOUBLE PRECISION statements, which must be changed to REAL * 8 for equivalent IBM function.

Finally, the main programs MAINT and BLIST must be changed. The program statements must be removed, and the IF EOF statements must be replaced with the IBM option END = NNN for read statements.

APPENDIX

REPLACEMENT DATA BANK ROUTINES FOR

IBM SYSTEM OPERATION

```
C
      SUBROUTINE DBLOAD
C       LOAD THE BANK FROM SEQUENTIAL FILE
      DIMENSION B(40),NA(40),C(40),NC(40),NAME(2),D(1000)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1)),(C(1),NC(1))
      CALL DBSTRT
C         FIND OUT HOW MANY SETS ARE IN THE BANK
      CALL DBREAD(1,NSETS)
      WRITE(6,40) NSETS
   40 FORMAT('1',I5,' SETS BEING LOADED IN BANK')
C         LOOP THROUGH THE SETS
      IF (NSETS.EQ.0) GO TO 550
      DO 500 ISET=1,NSETS
      CALL DBREAD(2,NAME)
      CALL DBREAD(1,NVAR)
      WRITE(6,50) NAME,NVAR
   50 FORMAT (' SET ',2A4,' HAS',I5,' VARIABLES')
      CALL DBSET(NAME,1)
      DO 400 IVAR=1,NVAR
      CALL DBREAD(1,IDNO)
      CALL DBREAD(1,IPI)
      CALL DBREAD(1,ISI)
      ISIZE=IPI*ISI
      IF (ISIZE.GT.1000) GO TO 150
      CALL DBREAD(ISIZE,D)
      CALL DBSTRC(IDNO,IPI,ISI,D)
      GO TO 400
  150 CALL DBSTOM(IDNO,IPI,ISI)
      DO 200 JSI=1,ISI
      IF (IPI.GT.1000) GO TO 170
      CALL DBREAD(IPI,D)
      CALL DBSTRP(IDNO,1,IPI,JSI,JSI,D,1000)
      GO TO 200
  170 DO 190 JPI=1,IPI,1000
      ILIM=MIN0(JPI+999,IPI)
      CALL DBREAD(ILIM+1-JPI,D)
      CALL DBSTRP(IDNO,JPI,ILIM,JSI,JSI,D,1000)
  190 CONTINUE
  200 CONTINUE
  400 CONTINUE
  500 CONTINUE
  550 CONTINUE
C         CLOSE OUT BUFFER
      CALL DBDONE
      WRITE(6,600)
  600 FORMAT ('0BANK LOAD COMPLETED')
      RETURN
      END
C
C
C
      SUBROUTINE DBSTRT
C        CALL DBSTRT TO INITIALIZE A NEW BANK
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      DEFINE FILE 29(15000, 40,U,IDUM)
      IERROR=0
      IERPRT=1
      DO 10 I=1,40
   10 NA(I)=0
      LREC=1
      NA( 38)=LREC
      CALL DBPUT(1,A)
      IA=0
```

```
      IB=0
      RETURN
      END
C
C
C
      SUBROUTINE DBOPEN
C         CALL DBOPEN TO OPEN AN EXISTING BANK
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      DEFINE FILE 29(15000, 40,U,IDUM)
      IERROR=0
      IERPRT=1
      IA=0
      IB=0
      IAS=0
      CALL DBGET(1,A)
      LREC=NA( 38)
      RETURN
      END
C
C
C
      SUBROUTINE DBSET(NAME,IOPT)
C         CALL SET TO DEFINE A DATA SET
C
C         IOPT VALUES AND MEANING
C           1    USE     IF FOUND.     CREATE IF NOT FOUND.
C           2    USE     IF FOUND.     ERROR  IF NOT FOUND.
C           3    ERROR   IF FOUND.     CREATE IF NOT FOUND.
C           4    DELETE  IF FOUND.     RETURN IF NOT FOUND.
C
      DIMENSION NAME(2)
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      IERROR=0
      IA=0
      IB=0
      ID=1
   10 CALL DBGET(ID,A)
      IDS=ID
      DO 100 I=1, 31,5
      IF (NA(I).EQ.0) GO TO 150
      IF (NA(I).NE.NAME(1)) GO TO 100
      IF (NA(I+1).EQ.NAME(2)) GO TO 200
  100 CONTINUE
      ID=NA( 40)
      IF (ID.NE.0) GO TO 10
C         ADD A NEW DIRECTORY ENTRY
      LREC=LREC+1
      NA( 40)=LREC
      CALL DBPUT(IDS,A)
      ID=LREC
      DO 90 I=1, 40
   90 NA(I)=0
      CALL DBPUT(ID,A)
      GO TO 10
  150 CONTINUE
      IF (IOPT.EQ.2) GO TO 1000
      IF (IOPT.EQ.4) GO TO 250
      NA(I)=NAME(1)
      NA(I+1)=NAME(2)
      LREC=LREC+1
```

```
      NA(I+4)=LREC
      CALL DBPUT(IO,A)
      IA=LREC
      IAS=IA
      NA(1)=NAME(1)
      NA(2)=NAME(2)
  155 DO 160 I=5, 40
  160 NA(I)=0
      NA( 40)=IA
      CALL DBPUT(IA,A)
      GO TO 250
  200 IF (IOPT.EQ.4) GO TO 300
      IA=NA(I+4)
      IAS=IA
      CALL DBGET(IA,A)
      IF (NA(1).NE.NAME(1)) GO TO 1050
      IF (NA(2).NE.NAME(2)) GO TO 1050
      IF (IOPT.EQ.3) GO TO 1100
      IF (IOPT.EQ.4) GO TO 155
  250 RETURN
  300 NA(I)=-1
      CALL DBPUT(IO,A)
      IA=0
      GO TO 250
 1000 IF (IERPRT.NE.0) WRITE (6,1010) NAME
 1010 FORMAT ('0DATA SET ',2A4,' NOT FOUND')
      IERROR=1
      RETURN
 1050 IF (IERPRT.NE.0) WRITE (6,1060)
 1060 FORMAT ('0***DATA BANK IS INCONSISTANT')
      IERROR=2
      CALL EXIT
 1100 IF (IERPRT.NE.0) WRITE (6,1110) NAME
 1110 FORMAT ('0DATA SET ',2A4,' ALREADY EXISTS')
      IERROR=3
      IA=0
      RETURN
      END
C
C
C
C
C
      SUBROUTINE DBDONE
C         CLOSE A BANK BEFORE END OF JOB OR OVERLAY
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1))
      IERROR=0
      IA=0
      CALL DBGET(1,A)
      NA( 38)=LREC
      CALL DBPUT(1,A)
      RETURN
      END
C
C
C
      SUBROUTINE DBDUMP
C         DUMP TO PRINT THE CONTENTS OF THE BANK
      DIMENSION F(4)
      DIMENSION B( 40),NA( 40)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),NA(1),B(1))
      IB=1
```

```
      WRITE (6,123)
  123 FORMAT ('1DUMP OF DATA BANK')
      CALL DBGET(1,A)
      LRECD=NA(38)
      LREC=MAX0(LREC,LRECD)
      DO 1000 IR=1,LREC
      WRITE (6,124) IR
  124 FORMAT ('0RECORD',I5)
      CALL DBGET(IR,A)
      DO 200 I=1, 40,4
      J=I
      IE=I+3
      JE=J+3
      CALL MVC(F,16,A(J))
      DO 100 K=1,16
      IF (CLC(F,K,'A',1,1)*CLC(F,K,'I',1,1).LE.0.) GO TO 100
      IF (CLC(F,K,'J',1,1)*CLC(F,K,'R',1,1).LE.0.) GO TO 100
      IF (CLC(F,K,'S',1,1)*CLC(F,K,'Z',1,1).LE.0.) GO TO 100
      IF (CLC(F,K,'0',1,1)*CLC(F,K,'9',1,1).LE.0.) GO TO 100
      IF (CLC(F,K,' ',1,1).EQ.0.) GO TO 100
   50 CALL MVC5(F,K,'.',1,1)
  100 CONTINUE
      WRITE (6,125) I,(NA(K),K=I,IE),F,(A(K),K=J,JE)
  125 FORMAT (I4,4I8,3X,4A4,3X,4G14.7)
  200 CONTINUE
      WRITE(1,124) IR
 1000 CONTINUE
      RETURN
      END
C
C
C
C
      SUBROUTINE DBGET(IREC,BUF)
C         GENERALIZED DIRECT ACCESS DISK READ FOR DATABANK
      DIMENSION BUF(40)
      READ (29'IREC) BUF
      RETURN
      END
C
C
C
      SUBROUTINE DBPUT(IREC,BUF)
C         GENERALIZED DIRECT ACCESS DISK WRITE FOR DATABANK
      DIMENSION BUF(40)
      WRITE (29'IREC) BUF
      RETURN
      END
C
C
C
      SUBROUTINE DBRUN(NRUN,ICODE)
C         SYNTHESIZE A DATASET NAME FOR A RUN NUMBER "RUN  NNN"
      DIMENSION NAME(2)
      DATA NAME /'RUN ','    '/
      CALL INCORE(NAME(2),4)
      WRITE(99,99) NRUN
   99 FORMAT (I4)
      CALL DBSET(NAME,ICODE)
      RETURN
      END
C
C
      SUBROUTINE DBSAVE
C         STORE THE BANK ON A SEQUENTIAL FILE
```

```
      DIMENSION B(40),NA(40),C(40),NC(40),NAME(2),D(1000)
      COMMON /BUFR/ A( 40),LREC,IA,IB,IAS,IERROR,IERPRT
      EQUIVALENCE (A(1),B(1),NA(1)),(C(1),NC(1))
      REWIND 28
      CALL DBOPEN
C        FIND OUT HOW MANY SETS ARE IN THE BANK
      NSETS=0
      ID=1
   10 DO 100 I=1,31,5
      CALL DBGET(ID,C)
      IF (NC(I).EQ.0) GO TO 150
      IF (NC(I).EQ.-1) GO TO 100
C        SEE IF IT HAS ANY VARIABLES
      IC=NC(I+4)
      CALL DBGET(IC,C)
      IF (NC(5).EQ.0) GO TO 100
      NSETS=NSETS+1
  100 CONTINUE
      CALL DBGET(ID,C)
      IF (NC(40).EQ.0) GO TO 150
      ID=NC(40)
      GO TO 10
  150 WRITE(6,160) NSETS
  160 FORMAT('1',I5,' SETS BEING STORED FROM BANK')
C
      CALL DBSTOR(1,NSETS)
C
C        LOOP THROUGH THE SETS
      IF (NSETS.EQ.0) GO TO 550
      ID=1
  180 DO 500 I=1,31,5
      CALL DBGET(ID,C)
      IF (NC(I).EQ.0) GO TO 550
      IF (NC(I).EQ.-1) GO TO 500
      NAME(1)=NC(I)
      NAME(2)=NC(I+1)
C        COUNT THE VARIABLES
      NVAR=0
      IC=NC(I+4)
      ICS=IC
  220 CALL DBGET(IC,C)
      DO 240 J=5,33,4
      IF (NC(J).EQ.0) GO TO 250
      IF (NC(J+1)*NC(J+2).LE.0) GO TO 240
      NVAR=NVAR+1
  240 CONTINUE
      IC=NC(40)
      IF (IC.NE.ICS) GO TO 220
  250 IF (NVAR.EQ.0) GO TO 500
      WRITE(6,260) NAME,NVAR
  260 FORMAT (' SET ',2A4,' HAS',I5,' VARIABLES')
C
      CALL DBSTOR(2,NAME)
      CALL DBSTOR(1,NVAR)
C
C        LOOP THROUGH AGAIN STORING VARIABLES
      CALL DBSET(NAME,2)
      IC=ICS
  300 CALL DBGET(IC,C)
      DO 400 J=5,33,4
      IF (NC(J).EQ.0) GO TO 410
      IDNO=NC(J)
      IPI=NC(J+1)
      ISI=NC(J+2)
      IF (IPI*ISI.LE.0) GO TO 400
```

```
C
      CALL DBSTOR(1,IDNO)
      CALL DBSTOR(1,IPI)
      CALL DBSTOR(1,ISI)
C
      ISIZE=IPI*ISI
      IF (ISIZE.GT.1000) GO TO 310
      CALL DBFCHC(IDNO,IPI,ISI,D)
C
      CALL DBSTOR(ISIZE,D)
C
      GO TO 400
  310 DO 390 JSI=1,ISI
      IF (IPI.GT.1000) GO TO 320
      CALL DBFCHP(IDNO,1,IPI,JSI,JSI,D,1000)
C
      CALL DBSTOR(IPI,D)
C
      GO TO 390
  320 DO 350 JPI=1,IPI,1000
      ILIM=MIN0(JPI+999,IPI)
      CALL DBFCHP(IDNO,JPI,ILIM,JSI,JSI,D,1000)
C
      ISZ=ILIM+1-JPI
      CALL DBSTOR(ISZ,D)
C
  350 CONTINUE
  390 CONTINUE
  400 CONTINUE
      IC=NC(40)
      IF (IC.NE.ICS) GO TO 300
  410 CONTINUE
C        END OF VARIABLE LOOP
  500 CONTINUE
      CALL DBGET(ID,C)
      IF (NC(40).EQ.0) GO TO 550
      ID=NC(40)
      GO TO 180
C        END OF SET LOOP
  550 CONTINUE
C        CLOSE OUT BUFFER
      CALL DBSTOR(0,0)
      WRITE(6,600)
  600 FORMAT ('OBANK STORAGE COMPLETED')
      CALL DBDONE
      REWIND 28
      RETURN
      END
C
C
C
C
      SUBROUTINE DBSTOR(N,D)
C        MACHINE DEPENDENT ROUTINE FOR OUTPUT TO SEQUENTIAL FILE
      DIMENSION D(N),CARD(20)
      DATA IC /0/,IOUT /28/
      IF (N.EQ.0) GO TO 100
C        COPY D INTO CARD
      ID=0
   10 ID=ID+1
      IC=IC+1
      CARD(IC)=D(ID)
      IF (IC.LT.20) GO TO 50
      WRITE(IOUT,20) CARD
   20 FORMAT (20A4)
```

```
      IC=0
   50 IF (ID.EQ.N) RETURN
      GO TO 10
  100 IF (IC.EQ.0) RETURN
      ICP=IC+1
      DO 110 I=ICP,20
  110 CARD(I)=0
      WRITE(IOUT,20) CARD
      IC=0
      RETURN
      END
C
C
      SUBROUTINE DBREAD(N,D)
C        MACHINE DEPENDENT ROUTINE FOR INPUT FROM SEQUENTIAL FILE
      DIMENSION D(N),CARD(20)
      DATA IC /20/,IIN /27/
      ID=0
      IF (IC.NE.20) GO TO 50
   10 READ(IIN,20) CARD
   20 FORMAT (20A4)
      IC=0
   50 ID=ID+1
      IC=IC+1
      D(ID)=CARD(IC)
      IF (ID.EQ.N) RETURN
      IF (IC.EQ.20) GO TO 10
      GO TO 50
      END
C
C
C
C
      FUNCTION FGTNUM(CARD,NSTART,NEND,K,NCHAR)
      DIMENSION CARD(1)
      FGTNUM=0.
      CALL GETNUM(CARD,NSTART,NEND,XDUM,K,NCHAR)
      IF (NCHAR.NE.0) FGTNUM=XDUM
      RETURN
      END
C
C
C
      SUBROUTINE GETNUM(CARD,NSTART,NEND,XNUM,K,NCHAR)
      DIMENSION CARD(1)
      DIMENSION FIELD(5)
      REAL MINUS
      DATA BLANK/'    '/,Z/'Z   '/,DEC/'.   '/,PLUS1/'&   '/,
     1   PLUS2/'+   '/, MINUS/'-   '/, EQUAL1/'#   '/, EQUAL2/'=   '/,
     2   COMMA/',   '/,E/'E   '/,D/'D   '/
      NCHAR=0
      XNUM=0.0
      KL=NEND+1
      IF(NEND.LT.NSTART) GO TO 300
      N1=NSTART
      N2=NEND-1
      I=NSTART-1
      K=I+1
      IF(CLC(CARD,K,Z,1,1).GT.0.0.OR.CLC(CARD,K,DEC,1,1).EQ.0.0.OR.
     1  CLC(CARD,K,PLUS1,1,1).EQ.0.0.OR.CLC(CARD,K,PLUS2,1,1).EQ.0.0
     2.OR. CLC(CARD,K,MINUS,1,1).EQ.0.0) GO TO 100
      IF(N1.GT.N2) GO TO 300
      DO 50 I=N1,N2
      K=I+1
      IF((CLC(CARD,I,BLANK,1,1).EQ.0.0.OR.CLC(CARD,I,EQUAL1,1,1).EQ.0.0
```

```
      1.OR.CLC(CARD,I,EQUAL2,1,1).EQ.0.0.OR.CLC(CARD,I,COMMA,1,1).EQ.0.0)
      2 .AND.
      1 (CLC(CARD,K,Z,1,1).GT.0.0.OR.CLC(CARD,K,DEC,1,1).EQ.0.0.OR.
      2  CLC(CARD,K,PLUS1,1,1).EQ.0.0.OR.CLC(CARD,K,PLUS2,1,1).EQ.0.0
      3.OR. CLC(CARD,K,MINUS,1,1).EQ.0.0)) GO TO 100
   50 CONTINUE
      GO TO 300
  100 CONTINUE
      IDEC = 0
      IF(CLC(CARD,K,DEC,1,1).EQ.0.0) IDEC = 1
      IF(K.EQ.NEND) GO TO 200
      KP1 = K+1
      DO 150 I=KP1,NEND
      KL=I
      IF(CLC(CARD,I,Z,1,1).GT.0.0.OR.(CLC(CARD,I,DEC,1,1).EQ.0.0.AND.
     1  IDEC.EQ.0)) GO TO 125
      GO TO 200
  125 CONTINUE
      IF(CLC(CARD,I,DEC,1,1).EQ.0.0) IDEC = 1
  150 CONTINUE
      KL=KL+1
  200 CONTINUE
      NCHAR = MIN0(20,KL-K)
      DO 250 I=1,5
      FIELD(I) = BLANK
  250 CONTINUE
      CALL MVC5(FIELD,21-NCHAR,CARD,K,NCHAR)
      CALL INCORE(FIELD,20)
      READ(5,1000) XNUM
      IF(KL.GE.NEND) GO TO 300
      N1=KL
      N2=NEND-1
      DO 400 I=N1,N2
      IF(CLC(CARD,I,BLANK,1,1).NE.0.0) GO TO 410
  400 CONTINUE
      GO TO 300
  410 IF(CLC(CARD,I,E,1,1).NE.0.0.AND.CLC(CARD,I,D,1,1).NE.0.0)
     X GO TO 300
      IP1=I+1
      DO 420 K=IP1,NEND
      IF(CLC(CARD,K,BLANK,1,1).NE.0.0) GO TO 430
  420 CONTINUE
      GO TO 300
  430 IF(CLC(CARD,K,Z,1,1).LE.0.0.AND.CLC(CARD,K,PLUS1,1,1).NE.0.0.AND.
     X    CLC(CARD,K,PLUS2,1,1).NE.0.0.AND.CLC(CARD,K,MINUS,1,1).NE.0.0)
     X GO TO 300
      KL=K+1
      IF(K.EQ.NEND) GO TO 440
      KP1=K+1
      DO 450 I=KP1,NEND
      KL=I
      IF(CLC(CARD,I,Z,1,1).LE.0.0) GO TO 440
  450 CONTINUE
      KL=KL+1
  440 CONTINUE
      NCHAR1=MIN0(20,KL-K)
      DO 460 I=1,5
      FIELD(I)=BLANK
  460 CONTINUE
      CALL MVC5(FIELD,21-NCHAR1,CARD,KL   -NCHAR1,NCHAR1)
      CALL INCORE(FIELD,20)
      READ(5,1001) IEXP
      XNUM=XNUM*10.**IEXP
      NCHAR=NCHAR+NCHAR1+1
  300 K=KL
```

```
      RETURN
 1000 FORMAT(F20.0)
 1001 FORMAT(I20)
      END
C
C
      LOGICAL FUNCTION SCAN(CARD,WORD)
      LOGICAL*1 CARD(1),WORD(1),DOLAR
      LOGICAL COMPAR
      DATA DOLAR /'$'/
      SCAN=.FALSE.
      DO 10 LEN=1,80
      IF (CLC(WORD,LEN+1,DOLAR,1,1).EQ.0.) GO TO 20
   10 CONTINUE
      RETURN
   20 SCAN=COMPAR(CARD,1,80,LEN,WORD,J)
      RETURN
      END
C
C
      LOGICAL FUNCTION COMPAR(C,I,L,N,CHAR,J)
      LOGICAL *1 C(1),CHAR(1),BLK
      DATA BLK /' '/
      NUM=N
      DO 5 K=1,N
      IF(CLC(CHAR,K,BLK,1,1).EQ.0.) NUM=NUM-1
    5 CONTINUE
      IL=L-NUM+1
      IF(I.GT.IL) GO TO 10
      DO 15 K=I,IL
      IF(CLC(C,K,BLK,1,1).EQ.0.) GO TO 15
      K1=K
      K2=1
      IL1=IL-1
   20 IF(K2.GT.N) GO TO 25
      IF(CLC(CHAR,K2,BLK,1,1).NE.0.) GO TO 30
      K2=K2+1
      GO TO 20
   30 IL1=IL1+1
   40 IF(K1.GT.IL1) GO TO 15
      IF(CLC(C,K1,BLK,1,1).NE.0.) GO TO 45
      K1=K1+1
      GO TO 40
   45 IF(CLC(C,K1,CHAR,K2,1).NE.0.) GO TO 15
      K1=K1+1
      K2=K2+1
      GO TO 20
   15 CONTINUE
      GO TO 10
   25 J=K1
      COMPAR=.TRUE.
      RETURN
   10 J=I
      COMPAR=.FALSE.
      RETURN
      END
```


*U.S.Government Printing Office: 1981 — 559-003/2015